

AD-A252 633

AL-TR-1992-0057



**INFORMATION INTEGRATION FOR
CONCURRENT ENGINEERING (IICE)
IDEF3 PROCESS DESCRIPTION CAPTURE
METHOD REPORT**

**Richard J. Mayer, PhD
Thomas P. Cullinane, PhD
Paula S. deWitte, PhD
William B. Knappenberger
Benjamin Perakath
M. Sue Wells**

**DTIC
ELECTE
JUL 09 1992
S D**

**KNOWLEDGE BASED SYSTEMS, INCORPORATED
2726 LONGMIRE
COLLEGE STATION, TEXAS 77845**

**HUMAN RESOURCES DIRECTORATE
LOGISTICS RESEARCH DIVISION**

MAY 1992

INTERIM TECHNICAL REPORT FOR PERIOD FEBRUARY 1991 - APRIL 1992

Approved for public release; distribution is unlimited.

92-17884

92 7 08 067



**AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6573**

ARMSTRONG

LABORATORY

NOTICES


This technical report is published as received and has not been edited by the technical editing staff of the Armstrong Laboratory.

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation, or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

The Public Affairs Office has reviewed this report, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This report has been reviewed and is approved for publication.


MICHAEL K. PAINTER
Program Manager


BERTRAM W. CREAM, Chief
Logistics Research Division

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
<small>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.</small>				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE May 1992	3. REPORT TYPE AND DATES COVERED Interim - February 1991 to April 1992		
4. TITLE AND SUBTITLE IDEF3 Process Description Capture Method Report		5. FUNDING NUMBERS C F33615-90-C-0012 PE 63106F PR 2940 TA 01 WU 08		
6. AUTHOR(S) Richard J. Mayer, PhD Thomas Cullinane, PhD Paula S. deWitte, PhD		William B. Knappenberger Benjamin Perakath M. Sue Wells		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Knowledge Based Systems, Inc. 2726 Longmire College Station, TX 77845		8. PERFORMING ORGANIZATION REPORT NUMBER KBSI-IICE-90-STR- 01-0592-02		
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Armstrong Laboratory Human Resources Directorate Logistics Research Division Wright-Patterson AFB, OH 45433		10. SPONSORING / MONITORING AGENCY REPORT NUMBER AL-TR-1992-0057		
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for Public Release; distribution is unlimited		12b. DISTRIBUTION CODE A		
13. ABSTRACT (Maximum 200 words) This document provides a method overview, practice and use description, and language reference for the IDEF3 Process Description Capture Method. The name IDEF originates from the Air Force program for Integrated Computer-Aided Manufacturing (ICAM) from which the first ICAM Definition, or IDEF, methods emerged. It was in recognition of this foundational work, and in support of an overall strategy to provide a family of mutually-supportive methods for enterprise integration, that continued development of IDEF technology was undertaken. More recently, with their expanded focus and widespread use as part of Concurrent Engineering, Total Quality Management (TQM), and business re-engineering initiatives, the IDEF acronym has been re-cast as the name referring to an integrated family of Integration Definition methods. IDEF3 was designed as a complementary addition to the IDEF family of methods serving the role of a knowledge acquisition and requirements definition tool that structures the user's understanding of how a given process, event, or system works around process flow and object-state transition descriptions. A special-purpose graphical language accompanying the IDEF3 method serves to highlight temporal precedence and causality relationships relative to the process or event being described.				
14. SUBJECT TERMS process modeling, IDEF, method, methodology, knowledge acquisition, requirements definition, information systems, information engineering, systems engineering, integration			15. NUMBER OF PAGES 140	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

Contents

Figures	v
Foreword.....	ix
Method Anatomy.....	x
Family of Methods.....	xii
Preface	xv
Acronyms	xvii
1.0 Introduction.....	1
1.1 Motivation	2
1.1.1 Enhance the Productivity of Business Systems Analysis	2
1.1.2 Facilitate Design Data Life-cycle Management	2
1.1.3 Support the Project Management Process	3
1.1.4 Facilitate the System Requirements Definition Process	3
1.2 Potential Uses of IDEF3	4
1.3 Example of an IDEF3 Process Description	5
2.0 IDEF3 Overview.....	9
2.1 Scenarios and Objects: The Organizing Structure for IDEF3 Descriptions.....	9
2.2 Process-centered Views: The Process Flow Diagrams	11
2.3 Object-centered Views: The Object State Transition Network Diagrams	14
3.0 Basic Elements of IDEF3 Process Descriptions	19
3.1 Units of Behavior	21
3.1.1 Unit of Behavior Elaborations.....	22
3.1.2 Unit of Behavior Decompositions	25
3.1.3 Unit of Behavior Numbering Scheme	27
3.1.4 Partial Descriptions	28

3.2	Links	29
3.2.1	Link Types	30
3.2.2	Link Specification Documents	32
3.2.3	Link Numbers	34
3.3	Junctions	34
3.3.1	Junction Types	34
3.3.1.1	AND, XOR, and OR Junctions	35
3.3.1.2	Fan-in and Fan-out Junctions	35
3.3.1.3	Synchronous and Asynchronous Junctions	37
3.3.1.4	Junction Semantics	37
3.3.2	Junction Combinations	38
3.3.3	Junction Numbering Scheme	40
3.4	Referents	41
3.5	Combining the Basic Building Blocks	47
3.5.1	Units of Behavior and Link Combinations	47
3.5.2	Combining Units of Behavior with Links and Junctions	49
3.5.3	The Use of Referents in IDEF3 Diagrams	54
3.6	Object State Transition Network Descriptions	56
3.6.1	Object vs. Object State	58
3.6.2	OSTN Description Components	59
3.6.3	The Semantics and Use of Object State Transition Network Diagrams	62
4.0	Development of IDEF3 Descriptions	65
4.1	Bounding the Description Capture Project	66
4.1.1	Defining the Purpose	67
4.1.2	Determine Initial Scope and Level of Detail	68
4.1.3	Identify Major Organizational Scenarios	69
4.2	Collect Data	69
4.2.1	Identify Experts and Prepare for Data Collection	70
4.2.2	Interview Domain Experts	72
4.2.2.1	Collect Names of Objects	73

4.2.2.2	Collect Names of Activities and Causality Relations	73
4.2.2.3	Collect Situation Descriptions	73
4.2.2.4	Maintain Description Findings Pools.....	74
4.3	Formulate Process Flow Descriptions.....	74
4.3.1	Develop Unit of Behavior Elaborations and Link Specifications.....	77
4.3.2	Develop Decompositions for Selected Units of Behavior	77
4.3.3	Cross-reference with IDEFØ Models	78
4.4	Summarize Object State Transitions.....	79
4.4.1	Select Objects of Interest	79
4.4.2	Characterize Object State Transitions and Layout the Object State Transition Network Diagram	80
4.4.3	Cross-reference to IDEF1 Models	81
4.5	Validate IDEF3 Process Descriptions.....	81
4.5.1	Motivation.....	81
4.5.2	Build and Distribute Kits	82
4.5.2.1	Roles in the Kit Review Process	83
4.5.2.2	The IDEF3 Kit Review Cycle.....	83
4.5.2.3	Types of IDEF3 Kits.....	85
4.5.2.4	Guidelines for Analysts and Commentors	86
4.5.2.5	Contents of IDEF3 Kits.....	88
5.0	IDEF3 Development: Barbershop Example	97
5.1	Define Purpose and Scope	97
5.2	Collect Data.....	98
5.2.1	Interview Domain Expert and Acquire Initial Description....	98
5.2.2	Analyze Description for Data Identification.....	99
5.3	Formulate Process Flow Descriptions.....	101
5.3.1	Identify the Sequence of Units of Behavior in the Diagram... ..	101
5.3.2	Analyze Data for Unit of Behavior Elaborations	104
5.3.3	Review Process Flow Description(s) with Domain Experts	104

5.4	Formulate Object State Transition Network Diagrams	112
6.0	Understanding IDEF3 Process Descriptions	115
6.1.	Description Reading Steps	115
6.2	Quick Reading of IDEF3 Process Descriptions: An Example.....	117
6.2.1	The Big Picture	117
6.2.2	Scan the Diagram.....	117
6.2.3	Understand the Scenario	118
7.0	Practical Guidelines for Using the IDEF3 Method	121
7.1	How to Construct Valid IDEF3 Diagrams.....	121
7.1.1	Model Theoretic Validation Rules for IDEF3 Process Descriptions	122
7.2	Some Common Errors and Guidelines to Constructing IDEF3 Diagrams	124
7.2.1	Fan-out XOR Junction Followed by a Fan-in AND.....	124
7.2.2	Multiple Precedence Links Emerging from a Unit of Behavior Box	126
7.2.3	Multiple Leftmost Points for a Scenario or a Decomposition	127
	Bibliography	129
	Index	133

Accession For		
NTIS GRA&I	<input checked="" type="checkbox"/>	
DTIC TAB	<input type="checkbox"/>	
Unannounced	<input type="checkbox"/>	
Justification		
By		
Distribution/		
Availability Codes		
Dist	Avail and/or Special	
A-1		



Figures

Figure 1-1	IDEF3 Process Description Example: Painting a Part.....	5
Figure 1-2	Object State Transition Network Diagram for Paint	7
Figure 2-1	Example of a Process Flow Diagram	12
Figure 2-2	Example of a Decomposition	14
Figure 2-3	Object State Transition Network (OSTN) Diagram Concepts...	15
Figure 2-4	An Example Process Flow Description Diagram.....	16
Figure 2-5	Object State Transition Network Diagram for the Purchase Request Form.....	16
Figure 3-1	Symbols Used for IDEF3 Process Description Diagrams.....	20
Figure 3-2	Unit of Behavior Elaboration Document.....	23
Figure 3-3	Decomposition 3.1 of Receive and Activate Contract	26
Figure 3-4	Decomposition 10.1 of Hold Kick-off Meeting UOB.....	26
Figure 3-5	The Project Manger's View Decomposition.....	27
Figure 3-6	Unit of Behavior Numbering Scheme	28
Figure 3-7	Disconnected UOB Example	29
Figure 3-8	IDEF3 Link Types	30
Figure 3-9	Example of a Relational Link	31
Figure 3-10	Link Specification Document.....	33
Figure 3-11	Anatomy of a Junction	35
Figure 3-12	Fan-in Junctions and Their Semantics	36
Figure 3-13	Fan-out Junctions and Their Semantics	36
Figure 3-14	Asynchronous AND Junction Example	39
Figure 3-15	Synchronous AND Junction Example	39
Figure 3-16	Asynchronous OR Junction Example.....	40

Figure 3-17 Referent Symbol Structure	42
Figure 3-18 Referent Symbol Syntax.....	43
Figure 3-19 Junction Constraint Referent	46
Figure 3-20 Precedence Link.....	48
Figure 3-21 Dashed Link Example	49
Figure 3-22 Object Flow Link	49
Figure 3-23 Asynchronous AND Junctions	50
Figure 3-24 Synchronous AND Junctions	51
Figure 3-25 Asynchronous OR Junctions	52
Figure 3-26 Synchronous OR Junctions	52
Figure 3-27 Fan-out AND Followed by a Fan-in OR Junction.....	53
Figure 3-28 Object Referent Example	55
Figure 3-29 Referents in Process Flow Diagrams.....	55
Figure 3-30 Referents to Elaborations.....	56
Figure 3-31 Referent to an Elaboration.....	57
Figure 3-32 Symbols Used for Object State Transition Network Diagrams .	57
Figure 3-33 Object State Transition Diagram	58
Figure 3-34 Object State Transition Network Description Form	60
Figure 3-35 Object State Description Form	61
Figure 3-36 Example to Illustrate OSTN Concepts.....	63
Figure 4-1 IDEF3 Description Summary Form	67
Figure 4-2 Source Material Log	71
Figure 4-3 Example IDEF3 Object Pool	75
Figure 4-4 Unit of Behavior Fields	78

Figure 4-5	IDEF3 Kit Cycle	84
Figure 4-6	IDEF3 Kit Review Cover Sheet	90
Figure 4-7	IDEF3 Kit Contents Sheet	91
Figure 4-8	IDEF3 Kit Diagram Form.....	93
Figure 5-1	First Steps in Scenario Description.....	102
Figure 5-2	Diagram with First Path Complete.....	102
Figure 5-3	Diagram Near Completion	103
Figure 5-4	Complete Flow Diagram Before First Review.....	103
Figure 5-5	Elaborations.....	105
Figure 5-6	Elaborations (Continued).....	106
Figure 5-7	Final Process Flow Diagram	109
Figure 5-8	Elaboration for Customer Leaving Dissatisfied.....	110
Figure 5-9	Elaboration Attached to a Junction.....	111
Figure 5-10	Link Specification Document.....	112
Figure 5-11	Object State Transition Network Example.....	113
Figure 6-1	Example IDEF3 Diagram	118
Figure 6-2	Partition the Diagram.....	118
Figure 6-3	Analyzing the Structures	119
Figure 7-1	Structures	124
Figure 7-2	Invalid XOR/AND Structure Example	125
Figure 7-3	Example of an Ambiguous Branch in an IDEF3 Diagram.....	126
Figure 7-4	The Corrected Paint Shop IDEF3 Diagram	126
Figure 7-5	An IDEF3 Scenario With Multiple Leftmost Points.....	127
Figure 7-6	Correct IDEF3 Diagram for the Assembly Shop Scenario.....	127

Foreword

The Department of Defense (DoD) has long recognized the opportunity for significant technological, economic, and strategic benefits attainable through the effective capture, control, and management of information and knowledge resources. Like manpower, materials, and machines, information and knowledge assets are recognized as vital resources that can be leveraged to achieve competitive advantage. The Air Force Information Integration for Concurrent Engineering (IICE) program, sponsored by the Armstrong Laboratory's Logistic Research Division, was established as part of a commitment to further the development of technologies that will enable full exploitation of these resources.

The IICE program was chartered with developing the theoretical foundations, methods, and tools to successfully implement and evolve towards an information-integrated enterprise. These technologies are designed to leverage information and knowledge resources as the *key* enablers for high quality systems that achieve better performance in terms of both life-cycle cost and efficiency. The subject of this report is one of a family of methods that collectively constitute a technology for leveraging available information and knowledge assets. The name IDEF originates from the Air Force program for Integrated Computer-Aided Manufacturing (ICAM) from which the first ICAM Definition, or IDEF, methods emerged. It was in recognition of this foundational work, and in support of an overall strategy to provide a family of mutually-supportive methods for enterprise integration, that continued development of IDEF technology was undertaken. More recently, with their expanded focus and widespread use as part of Concurrent Engineering, Total Quality Management (TQM), and business re-engineering initiatives, the IDEF acronym has been re-cast as the name referring to an integrated family of Integration Definition methods. Before discussing the development strategy for providing an integrated family of IDEF methods,

however, the following paragraphs will briefly introduce what constitutes a method.

Method Anatomy

A **method** is an organized, single-purpose discipline or practice (Coleman, 1989). A method may have a formal theoretic foundation. However, most do not (except possibly in the eyes of the developer of the method). Generally, methods evolve as a distillation of *best-practice* experience in a particular domain of cognitive or physical activity. The term **methodology** has at least two common usages. The first use is to refer to a class of similar methods. So, one may hear reference to the *function modeling methodology* referring to methods such as IDEFØ¹ and LDFD.² In an other sense, the term **methodology** is used to refer to *a collection of methods and tools, the use of which is governed by a process superimposed on the whole* (Coleman, 1989). Thus, it is common to hear the criticism that a tool (or method) has no underlying methodology. Such a criticism is often leveled at a tool (or method) which has a graphical language but for which no procedure for the appropriate application of the language or use of the resulting models is provided. For simplicity, the term **tool** is used to refer to a software system designed to support the application of a method.

Though a method may be thought of informally as simply a procedure for performing a task plus perhaps a representational notation, it may be described more formally as consisting of three components as illustrated in Figure F-1. Each method has (a) a definition, (b) a discipline, and (c) many uses. The definition specifies the basic intuitions and motivation behind the method, the concepts involved, and the theory of its operation. The discipline includes the procedure by which the method is applied and the language, or syntax, of the method. The procedure associated with the method discipline provides the practitioner with a reliable process for achieving consistently

¹ICAM Definition method for Function Modeling

² Logical Data Flow Diagramming method

good results. The method syntax is provided to eliminate ambiguity among those involved in the development of complex engineering products. Many system analysis and engineering methods use a graphical syntax to provide visualization of collected data in such a way that key information can be easily extracted.³ The third element of the method anatomy, the use component, focuses on the context-specific application of the method.

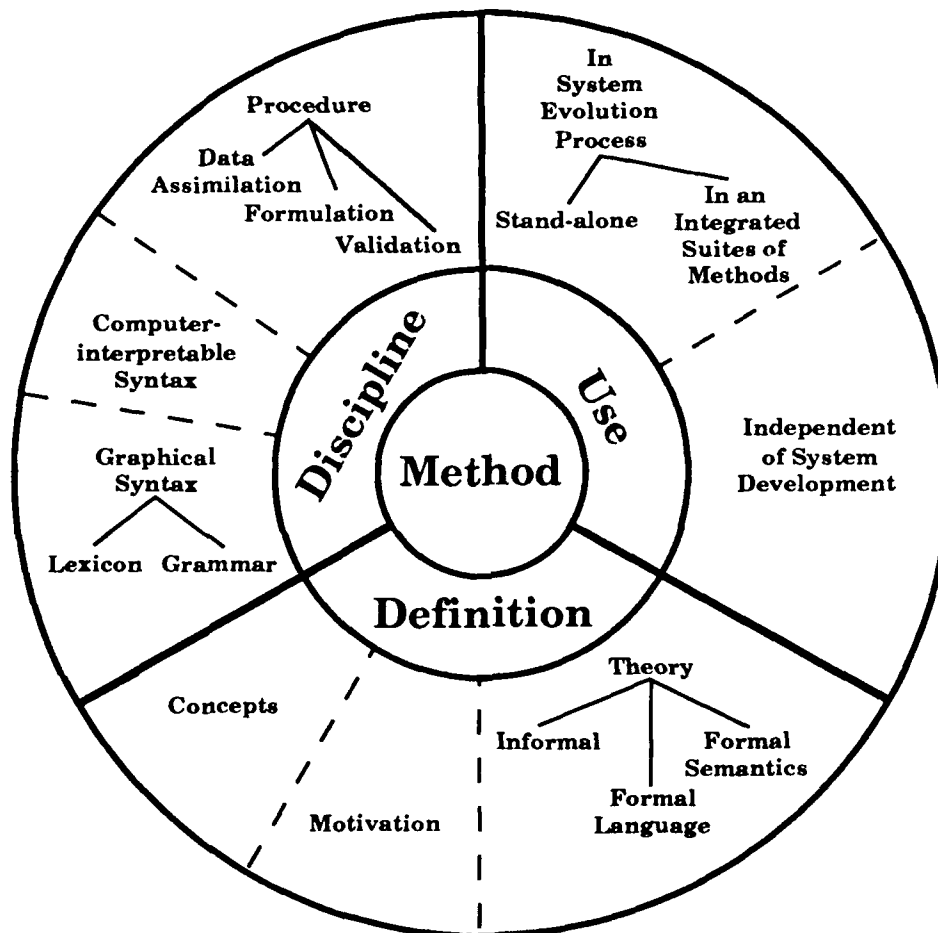


Figure F-1
Anatomy of a Method

³ Graphical facilities provided by a method language serve not only to document the analysis or design process undertaken, but more importantly, to highlight important decisions or relationships that must be considered during method application. The uniformities to which an expert becomes attuned over many years of experience are thus formally encoded in visualizations that emulate expert sensitivities.

Ultimately, methods are designed to facilitate a scientific approach to problem solving. This goal is accomplished by first, helping one understand the important objects, relations, and constraints that must be discovered, considered, or decided on; and second, by guiding the method practitioner through a disciplined approach, consistent with good-practice experience, towards the desired result. Formal methods, then, are specifically designed to raise the performance level (quality and productivity) of the novice practitioner to something comparable with that of an expert (Mayer, 1987).

Family of Methods

As Mr. John Zachman, in his seminal work on information systems architecture observed, "...there is not an architecture, but a set of architectural representations. One is not right and another wrong. The architectures are different. They are additive, complementary. There are reasons for electing to expend the resources for developing each architectural representation. And, there are risks associated with not developing any one of the architectural representations." Consistent, reliable creation of correct architectural representations, whether they be artificial approximations of a system (models) or purely descriptive representations, requires the use of a guiding method. These observations underscore the need for many "architectural representations," and correspondingly many methods.

Methods, and their associated architectural representations, focus on a limited set of system characteristics and explicitly ignore those that are not directly pertinent to the task at hand. Methods were never intended to evaluate and represent every possible state or behavioral characteristic of the system under study. If such a goal were achievable, the exercise would itself constitute building the actual system, thus negating the benefits to be gained through method application (e.g., problem simplification, low cost, rapid evaluation of anticipated performance, etc.).

The search for a single method, or modeling language, to represent all relevant system life cycle and behavioral characteristics, therefore, would

necessitate skipping the design process altogether. Similarly, the search for a single method to facilitate conceptualization, system analysis, and design continues to frustrate those making the attempt.

Recognizably, the plethora of special-purpose methods which typically provide few, if any, explicit mechanisms for integration with other methods, is equally frustrating. The IDEF family of methods is intended to strike a favorable balance between special-purpose methods whose effective application is limited to specific problem types, and "super methods" which attempt to include all that could ever be needed. This balance is maintained within the IDEF family of methods by providing explicit mechanisms for integrating the results of individual method application.

Critical method needs identified through previous studies and research and development activities⁴ have given rise to renewed effort in IDEF method integration and development activities, with an explicit mandate for compatibility among the family of IDEF methods. Providing for known method needs with a family of IDEF methods was not, however, the principal goal of methods engineering activity within the IICE program. The primary emphasis for these efforts was directed towards establishing the foundations for an engineering discipline guiding the appropriate selection, use, extension, and creation of methods that support integrated systems development in a cost-effective and reliable manner.

New methods development has struck out where known and obvious method voids existed (rather than re-inventing existing, and often very good methods) with the explicit mission to forge integration links with and between existing IDEF methods. When applied in a stand-alone fashion, IDEF methods serve to embody knowledge of good practice for the targeted fact collection,

⁴ Of particular note is the Knowledge-Based Integrated Information Systems Engineering (KBIISE) Project conducted at the Massachusetts Institute of Technology (MIT) in 1987 where a collection of highly qualified experts from academic and research organizations, government agencies, computer companies, and other corporations identified method and tool needs for large-scale, heterogeneous, distributed systems integration. See Defense Technical Information Center (DTIC) reports A195851 and A195857.

analysis, design, or fabrication activity. As with any good method, the IDEF methods are designed to raise the performance level of novice practitioners to a level that is comparable to that of an expert by focusing attention on important decisions while masking out irrelevant information and unneeded complexity. Viewed collectively as a complementary toolbox of methods technology, the IDEF family is designed to promote integration of effort in an environment where global competitiveness has become increasingly dependent upon the effective capture, management, and use of enterprise information and knowledge assets.

Preface

This document provides a method overview, practice and use description, and language reference for the IDEF3 Process Description Capture Method developed under the Information Integration for Concurrent Engineering (IICE) project, F33615-90-C-0012, funded by Armstrong Laboratory, Logistics Research Division, Wright-Patterson Air Force Base, Ohio 45433, under the technical direction of United States Air Force Captain Michael K. Painter. The prime contractor for IICE is Knowledge Based Systems, Inc. (KBSI), College Station, Texas. Dr. Paula S. deWitte is IICE Project Manager at KBSI, Dr. Richard J. Mayer is Principal Investigator, and Arthur A. Keen is Methods Engineering Thrust Manager.

The document is divided into the following seven sections:

1. Introduction
2. IDEF3 Overview
3. Basic Elements of IDEF3 Process Descriptions
4. Development of IDEF3 Process Descriptions
5. IDEF3 Development: Barber Shop Example
6. Understanding IDEF3 Process Descriptions
7. Practical Guidelines for Using the IDEF3 Method

The introduction describes the motivations and potential uses for the IDEF3 method. A brief method overview is presented in Section 2.0. Section 3.0 provides a detailed description of the basic building blocks used to develop IDEF3 process flow descriptions. Sections 4.0 and 6.0 offer practical guidelines to both novice and experienced IDEF3 users for the systematic application of the method. Use of the method is demonstrated through a detailed example described in Section 5.0. Finally, Section 7.0 presents a few

tips and traps; awareness of these can aid in the effective use of the IDEF3 method.

The authors anticipate the use of this document for a wide variety of purposes. Thus, the material is presented in a manner that allows readers to obtain the needed knowledge without having to read the entire document. The following guidelines are suggested for the use of this document.

1. For an *executive overview*, read Sections 1.0 and 2.0.
2. To become *proficient* in the development of accurate IDEF3 process flow descriptions should read the entire manual. Place special emphasis on Sections 2.0, 3.0, 5.0, and 7.0.
3. *Experienced* IDEF3 analysts can use Sections 2.0, 3.0, and 7.0 as language references.
4. To become proficient in *reviewing* IDEF3 process flow descriptions, read Section 6.0 in detail and browse Sections 2.0 and 7.0.
5. An IDEF3 *project leader* should study Section 4.0 in detail, but must also have an understanding of the method in its entirety.

IDEF3 is designed to support the capture and structuring of *descriptions* of how a system works. IDEF3 development was motivated by the need to capture assertions made by knowledgeable experts about the behavior of a system in contrast to constructing engineering *models* that approximate system behavior. The ability to support the capture of real-world descriptions that are *partial* (incomplete) distinguishes IDEF3 from traditional process modeling methods.

KBSI acknowledges the technical input to this document made by previous work under the Integrated Information Systems Evolutionary Environment (IISEE) project performed by the Knowledge Based Systems Laboratory, Department of Industrial Engineering, Texas A&M University (Mayer, 1991).

Acronyms

FDR	Flight Discrepancy Report	12
GPSS	General-Purpose Simulation System	1
ID	Identifier	47
OSD	Object State Description Form	17
OSTN	Object State Transition Network	4
PFN	Process Flow Network.....	11
PMAA	Perform Mission Area Analysis.....	49
SIMAN	Simulation Analysis	1
SLAM	Simulation Language for Alternative Modeling.....	1
UOB	Unit of Behavior	6

1.0 Introduction

One of the most common communication mechanisms to describe a situation or process is a story told as an ordered sequence of events or activities. For example, an engineer often describes the design process of his company by telling a story about a product that was recently developed. Likewise, a shop floor supervisor may describe the operation of his manufacturing system by describing the process of building a product in his shop. IDEF3 was created specifically to capture descriptions of sequences of activities. Thus, the primary goal of IDEF3 is to provide a structured method by which a domain expert can express his knowledge of the operation of a particular system or organization. Knowledge acquisition in this method is enabled by the direct capture of assertions about real-world processes and events in a form that is most natural for capture. This includes the capture of assertions about the objects that participate in the process, assertions about supporting objects, and the precedence and causality relations between processes and events within the environment.

IDEF3 can be distinguished from other process modeling methods because it facilitates the capture of the *description* of what a system actually does. The IDEF2 Simulation Modeling Method and a host of other simulation languages (e.g., SIMAN, SLAM, GPSS, etc.), on the other hand, enable the development of mathematical idealizations, or *models*, that predict what a system will do. The implied difference between descriptions and models, though subtle, is an important one. A description is a recorded collection of assertions (statements, observations, or beliefs) which are held to be true by participants in a domain. These assertions are typically incomplete and possibly inaccurate with respect to how things actually occur within that domain. Models, on the other hand, are *idealizations* intended to represent certain relevant aspects of a system for purposes of prediction or analysis. They are thus assumed to be complete and accurate. Description capture is attractive as a strategy for knowledge acquisition when compared to model

building for several reasons. First, domain experts generally require less training to produce descriptions of their domain as opposed to developing models for their domain. Second, a description of a given situation is generally of higher utility than a model, since the description can easily be reused for a variety of purposes, including model building. In the past, a disadvantage of descriptions has been the lack of an effective means for organizing, displaying, and analyzing them. IDEF3 is a description organizing and capture method that addresses these needs.

1.1 Motivation

A primary motivation for the development of the IDEF3 method was to address the needs of business and industry in specific areas. Some of the more prominent motivations are described in the following sections.

1.1.1 Enhance the Productivity of Business Systems Analysis

One major motivation behind IDEF3 development was the perceived need to speed up the process of business systems modeling. In business re-engineering situations, systems analysis activities often start with the acquisition of an accurate description of the problem situation. Domain experts express their problems in terms of an ordered sequence of events or activities. Moreover, the specific ways in which activities and the objects that participate in them are related is generally described. Thus, to facilitate these activities, there is a need for both a *method* to facilitate the capture of the dynamics of business activities and process descriptions, and for a *representation medium* to store and manipulate this captured knowledge. IDEF3 fulfills these requirements by providing a structured approach to communicate such process information described by domain experts.

1.1.2 Facilitate Design Data Life-cycle Management

There is an identified need (Mayer, 1987) for a method to describe engineering design-data life cycles. To describe the design-data life cycle, it is

necessary to describe: 1) the artifacts or objects of design information (i.e., drawings, CAD models, etc.), 2) the state transitions through which these artifacts proceed, and 3) the decision logic or processes that determine the state transitions. IDEF3 provides mechanisms to describe this data life cycle information through the use of state transition diagrams.

1.1.3 Support the Project Management Process

Project management techniques are used to monitor and control projects in a wide variety of application domains. Several software tools have been developed to support these project management techniques. However, since these management techniques are modeling methods rather than description capture methods, they are unable to capture the complexities that occur in real project management situations. IDEF3 provides mechanisms to capture the constraints (including resource and temporal relationships) between the activities of a project. The IDEF3 language also provides the means to represent detailed information about the objects that participate in or are produced or used by the project activities. Furthermore, the activation of IDEF3 diagrams, which can be supported by an automated tool, will provide the means to monitor and control project activities in real-time.

1.1.4 Facilitate the System Requirements Definition Process

Another motivation for the development of IDEF3 was to provide the concepts, syntax, and procedures for building *system requirements descriptions*. These descriptions must be adequately detailed to determine if a delivered system is acceptable. This implies that the IDEF3 method must support descriptions of the following items.

1. Scenarios of organizational activities.
2. Roles of user types in these organizational activities.
3. User scenarios or user interaction with the information system at the user-function level.

4. System response to user functions.
5. User classes and delineation of user classes.
6. Declaration of timing, sequencing, and resource constraints.
7. User interface objects (e.g., menus, keywords, screens, and displays).

1.2 Potential Uses of IDEF3

The IDEF3 process flow description diagrams and the Object State Transition Network (OSTN) diagrams provide powerful mechanisms for data collection and analysis. An IDEF3 process flow description can be used to simplify and provide the data for many different purposes, including the following.

1. To provide a systematic method for recording and analyzing the raw data that results from fact-finding interviews in a systems analysis project.
2. To determine the impact of an organization's information resource on the major operating scenarios of an enterprise.
3. To provide a mechanism for documenting the decision procedures affecting the states and life cycle of critical shared data (particularly manufacturing, engineering, maintenance, and product definition data).
4. To define data configuration management and change control policy definition.
5. To support system design and design tradeoff analysis.
6. To provide powerful mechanisms to support the generation of simulation models.
7. To provide useful information for the creation of functional (IDEFØ) models.
8. To facilitate process mapping for the design of software to achieve real-time control by providing a mechanism for clearly defining the facts, decision points, and job classifications.

9. To provide an analyst with a method to clearly define the data needed to develop needs and requirements from a user viewpoint.
10. To collect and express the views of domain experts required for the development of expert systems.

1.3 Example of an IDEF3 Process Description

The following example illustrates how the basic building blocks of the IDEF3 method can be utilized to describe a typical manufacturing situation. Consider a workshop that paints a manufactured part which is subsequently used in the grouping of some heavy construction equipment. When asked to describe the painting process, the shop supervisor relates the following story.

Parts enter the shop ready for the primer coat to be applied. We apply a very heavy coat of primer by spraying paint in liquid form under high pressure. The paint is allowed to dry in a bake oven after which a paint coverage test is performed on the part. If the test reveals that not enough primer paint has been sprayed on the surface of the part, the part is rerouted through the paint shop. If the part passes the inspection, it is routed to the next stop in the manufacturing process where it is polished.

Figure 1-1 shows the IDEF3 *process flow description diagram* of this situation.

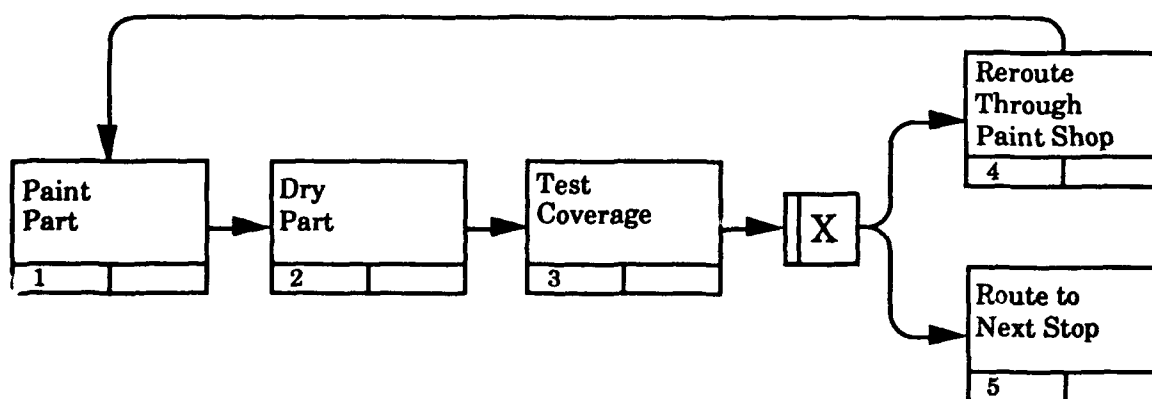


Figure 1-1
IDEF3 Process Description Example: Painting a Part

The processes described in the painting process description are clearly identified in the diagram and represented as labeled boxes numbered 1 through 5. Each box represents distinguishable packets of information about an event, decision, act, or process. That is, boxes represent types of happenings. Such happenings are referred to by the neutral term *units of behavior* (UOBs). The arrows (called *links*) connecting the boxes indicate the precedence relationships (or more generally *constraints*) that hold between the processes being described. The small box containing the "X" denotes a *junction*. A junction is a point in the process flow where a process flow path branches into multiple paths, or multiple process flow paths merge into one. Junctions describe the *flow logic* of the process. The process flow diagram in Figure 1-1 thus represents "The Process of Painting a Part" scenario. In IDEF3, scenarios bound the context of descriptions and are convenient artifacts for describing similar situations from different perspectives.

The IDEF3 diagram in Figure 1-1 represents a *process-centered view* of the paint shop. This view focuses on the assertions about the processes that occur and their ordering. Sometimes it is convenient to organize the description of a situation from an *object-centered view* (i.e., a participating object is the focus of attention). For this example, the paint could be considered an object that changes its state during the processes described in the shop. IDEF3 facilitates object-centered views through OSTN diagrams.

The OSTN diagram in Figure 1-2 is a graphical description of what happens to the paint within the paint shop described earlier. The labeled circles represent distinct *states* in which the paint can exist. Each arc (arrow) connecting the circles symbolizes a *state transition* (i.e., the activity of changing from one state to another). The banded boxes linked to the arrows (called *referents*) are aids to describe what can happen or must happen during the transition of an object from one state to another. For example, during the transition of the object *paint* from its liquid state within the paint machine to a solid state on the painted part, the processes represented by the UOB *Paint Part* and UOB *Dry Part* must both complete in the order that their referents are attached to the arc. The state *Paint Covered by New Layer* is

reached when the part is rejected and repainted with another layer. Parts which pass inspection are polished as indicated by *Paint Covered with Polish*. Neither description mentions movement of the part from one location to another. This is simply because the original dialog contained no information about such a movement. This is a key point in the use of IDEF3. IDEF3 is intended as a mechanism for structuring the assertions made by the domain expert. It does not force the completion of partial information with *modeling assumptions*.

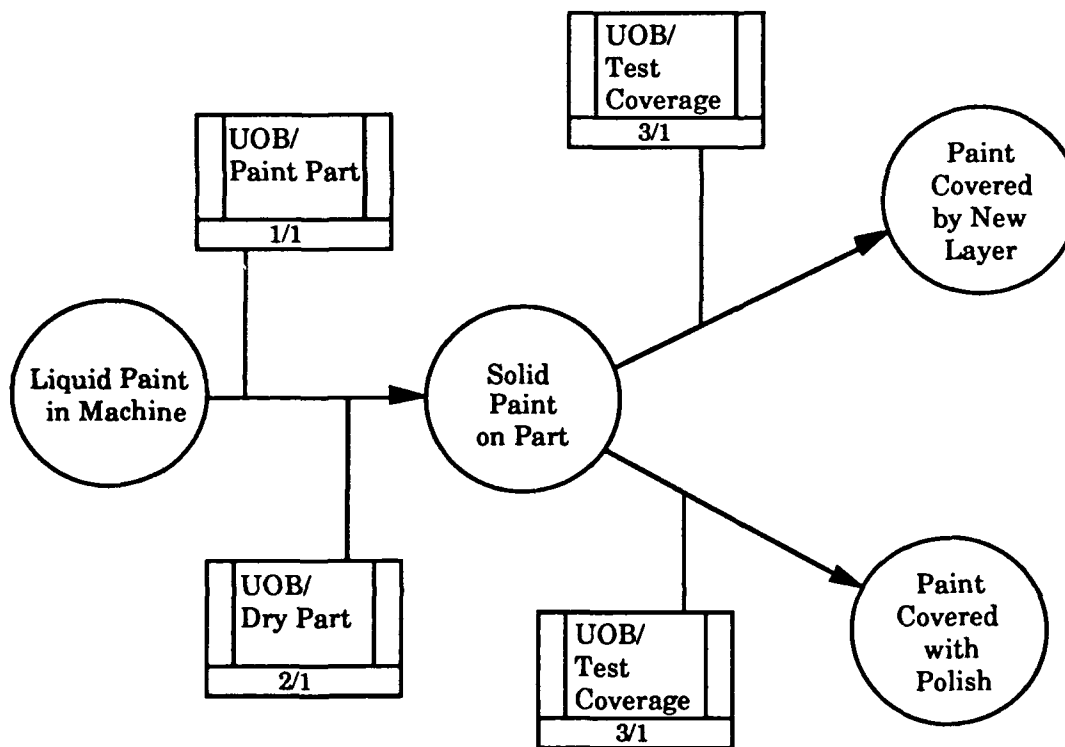


Figure 1-2
Object State Transition Network Diagram for Paint

2.0 IDEF3 Overview

This section provides a broad overview and examples of the descriptive organizing concepts of the IDEF3 method. Since any discussion of the organizing structures requires references to the basic IDEF3 elements,¹ these will be referred to but not fully defined until Section 3.0. An IDEF3 description is structured along two dimensions: the scenario dimension and the object dimension.

2.1 Scenarios and Objects: The Organizing Structure for IDEF3 Descriptions

The IDEF3 method uses a knowledge acquisition strategy centered on the capture of descriptions of process flow (processes and their temporal, causal, and logical relations) along with the identification of objects that participate in these processes and the state transitions of those objects. IDEF3 uses the notion of a *scenario* or story as the basic organizing structure for establishing the focus and boundary conditions of the process description. This feature exploits the tendency of humans to describe what they know in terms of an ordered sequence of activities that they have experienced or observed within the context of a given scenario or situation. A scenario can be thought of as: 1) a particular recurring situation within an organization for which documentation is required, 2) a set of situations that describe a typical class of problems addressed by an organization or system, or 3) the setting within which a process occurs. In IDEF3, scenarios serve as vehicles to organize collections of process-centered knowledge.

Since the primary role of a scenario is to bind the context of a process description, it is important to name it appropriately. Scenario names are

¹ IDEF3 elements are the basic language constructs of IDEF3, including UOBs, junctions, links, and referents.

often action verbs, gerunds, or verb phrases. A well-chosen scenario name will ensure that the users of the description make the appropriate associations with the real-world situations being described. The following examples are typical process flow scenario names.

1. Develop Die Design for Side Aperture Panel
2. Processing a Customer Complaint
3. Implement Engineering Change Request

Identifying, characterizing, and naming scenarios is a necessary step in the creation of IDEF3 descriptions.

IDEF3 uses the notion of an *object* as the basic organizing structure for establishing the focus for the object state transition description. An object in the IDEF3 method is any physical or conceptual thing that is recognized and referred to by participants in the domain as a part of their descriptions of what happens in their domain. Identifying, characterizing, and naming objects is also a necessary step in the creation of IDEF3 descriptions.

The next step is to use the basic elements of the IDEF3 language to express the assertions that will form the description. IDEF3 provides two different strategies for developing descriptions: 1) the process flow description strategy (which facilitates a process centered approach) and 2) the object state transition description strategy (which facilitates an object centered approach). An IDEF3 description may contain many process flow descriptions and many object state transition descriptions. The scenario concept is used to organize the process-centered views; the object concept is used to organize the object-state-transition-centered views. The collection of these organizing units and their contents *is* the IDEF3 description.

In summary, every IDEF3 description has associated with it one or more scenarios and one or more objects. These scenarios and objects define or bound the context of the entire description. The scenarios and objects are considered part of the description and are the organizing and scoping mechanism for the description. That is, recording that a particular named

physical or conceptual object is recognized by participants in a domain is considered part of the description of that domain. Thus, an object may not have an OSTN diagram associated with it in a description. Similarly, a scenario may or may not have a process flow diagram associated with it (i.e., its description may not yet be detailed). Yet these objects and scenarios are considered part of the description. The following two sections briefly introduce the description representation concepts and syntax available in the Process Flow Network (PFN) and OSTN of IDEF3.

2.2 Process-centered Views: The Process Flow Diagrams

The IDEF3 PFNs are the primary means for capturing, managing, and displaying process-centered knowledge. The display of a PFN is a process flow diagram. These diagrams provide a graphical medium that supports domain experts and analysts from a variety of application areas in communicating knowledge (complete or partial) about processes. This includes knowledge about events and activities, the objects that participate in those occurrences, and the constraining relations that govern the behavior of an occurrence.

A process-centered description is constructed in a systematic manner using the basic building blocks which are linked together in different ways. These building blocks have specific semantics associated with them. That is, they are used to represent certain kinds of activities or relations in the real-world. A detailed specification of these building blocks is given in Section 3.0. In Section 2.2, some of the important building blocks are explained, as well as how they are used to develop IDEF3 process flow descriptions.

The process flow diagram shown in Figure 2-1 depicts an aircraft maintenance process (more specifically, the processes associated with the management of a flight discrepancy). The labeled boxes with numbers are the UOBs associated with this scenario. Each UOB box represents a real-world process. The information recorded about a UOB includes 1) a name (often verb-based) that is indicative of what the UOB represents, 2) the

names of the objects that participate in the process and their properties, and 3) the relations that hold between the objects. The arrows between the UOBs are called *precedence links*; these depict the temporal precedence between the processes. Thus, the UOB at the source of a link would complete before a UOB at the end of the same link can start. For example, referring to Figure 2-1, the UOB labeled *Initiate FDR* (Flight Discrepancy Record) would need to complete before the UOB *Distribute FDR* can start.

In Figure 2-1, the boxes with a band on the left are called *junctions*. Junctions indicate either a split or a join in two or more process flows; essentially, they are used to capture the flow logic in processes which have multiple streams of flow. The labeled boxes without numbers are called *referents*. They act as labeled pointers to indicate some information detailed elsewhere in the IDEF3 process flow description. Referents point to other IDEF3 elements such as UOBs, scenarios, or objects.

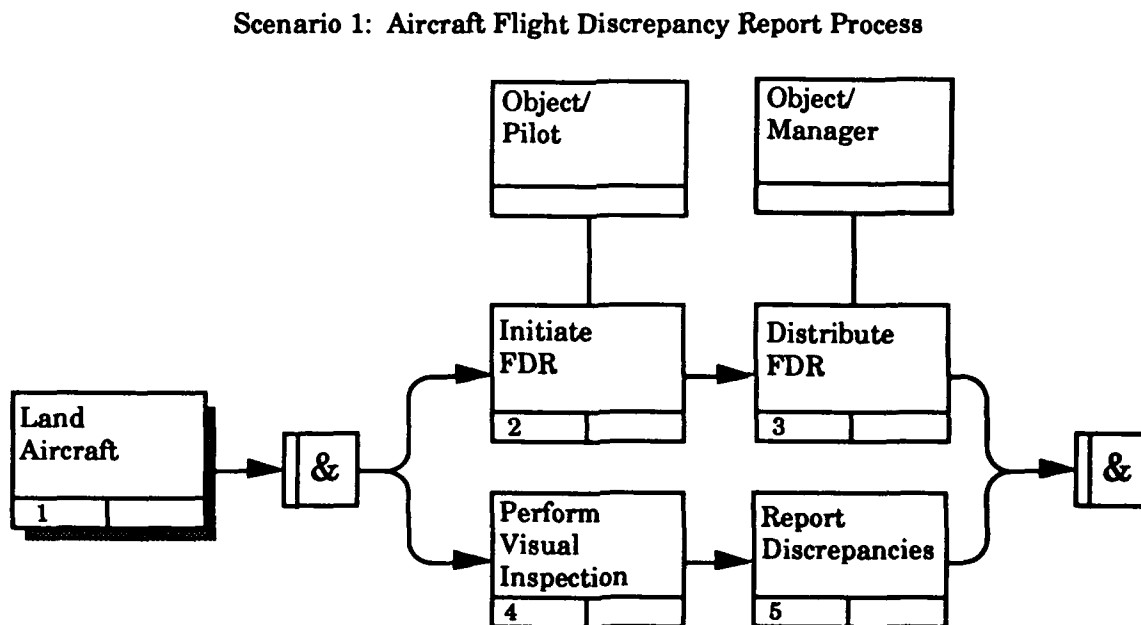
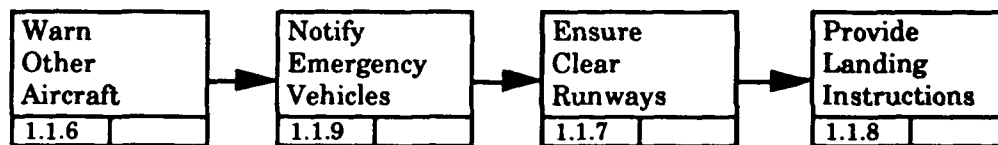


Figure 2-1
Example of a Process Flow Diagram

The IDEF3 diagram shown in Figure 2-1 depicts the activities that occur after a damaged aircraft lands. It can be interpreted in the following manner. Once the aircraft has landed, two courses of action are initiated simultaneously (the & symbol within the junction is similar to the logical operator AND). One of the activity sequences leads to the generation of an FDR. Visual inspection activities are initiated in parallel, resulting in reporting visual discrepancies. The maintenance activities terminate after *both* the FDR reports and the visual discrepancy reports have completed.

Two referents are used in the IDEF3 process flow diagram illustrated in Figure 2-1. The referent labeled *Pilot* indicates an object that is critical in the completion of the process *Initiate FDR*. The labeled referencing method used here highlights important information in the description. More detailed information about how the pilot participates in the FDR report generation would be contained in the elaboration of UOB *Initiate FDR*. In addition, there may also be an OSTN diagram for such a distinguished object. The use of the first junction with the & symbol in this IDEF3 diagram indicates the logic of the flow. That is, after the aircraft has landed, the processes *Initiate FDR* and *Perform Visual Inspection* will both be initiated. The rightmost & junction in Figure 2-1 indicates that both *Distribute FDR* and *Report Discrepancies* must complete before any additional processes can initiate.

The IDEF3 method provides the facility to capture descriptions at varying levels of abstraction by providing a mechanism called a *decomposition*. A decomposition provides a means of organizing a more detailed description of a UOB. A decomposition takes the form of another process flow diagram. The process flow diagram of a decomposition follows the same syntactic rules as those for a scenario and is created using the same IDEF3 elements. A UOB can have any number of different decompositions, all on the same level. The use of more than one decomposition for the same UOB is for the purpose of representing different points of view or providing greater details of the processing relating to the UOB. The UOB *Land Aircraft* in Figure 2-1 has one or more decomposition(s) attached to it, as indicated by its shadowed box. The process flow diagram of one such decomposition is shown in Figure 2-2.



Aircraft Controller View of Landing Aircraft (Decomposition 1.1)

Figure 2-2
Example of a Decomposition

The process description depicted in Figure 2-2 shows of the aircraft landing process from a particular point of view—that of an air-traffic controller. It is possible to conceive of other views for this process; for example, that of the pilot of the aircraft. Each view to be described would be presented in a separate decomposition with a unique label and number.

2.3 Object-centered Views: The Object State Transition Network Diagrams

IDEF3 OSTNs are the primary means provided by IDEF3 for capturing, managing, and displaying *object-centered* knowledge. The display of an OSTN is called an OSTN diagram. Such views cut across the PFNs and enable descriptions of objects which evolve through a number of states. OSTN diagrams provide a characterization of alternative states of an object. These diagrams allow the specification of the rules that govern the transitions that can take place between object states. Figure 2-3 illustrates some of the concepts used in OSTN diagrams. In these diagrams, labeled circles represent object states, arcs represent allowable transitions between states. The entry conditions, state descriptions, and exit conditions are actually recorded on a special form.

Each OSTN diagram focuses on one object. One of the first steps in the development of an OSTN is to identify all possible states in which the object can exist. Though a real-world object often evolves through a continuum of states, an OSTN diagram focus on those distinguished states that are of

particular interest to the domain expert. For each of these states, the OSTN diagram supports the specification of: 1) the conditions which characterize the state, 2) the conditions that will permit a transition *into* the state (*entry conditions*), and 3) the conditions that need to hold for the object to transition *out* of the state, (*exit conditions*) as shown in Figure 2-3.

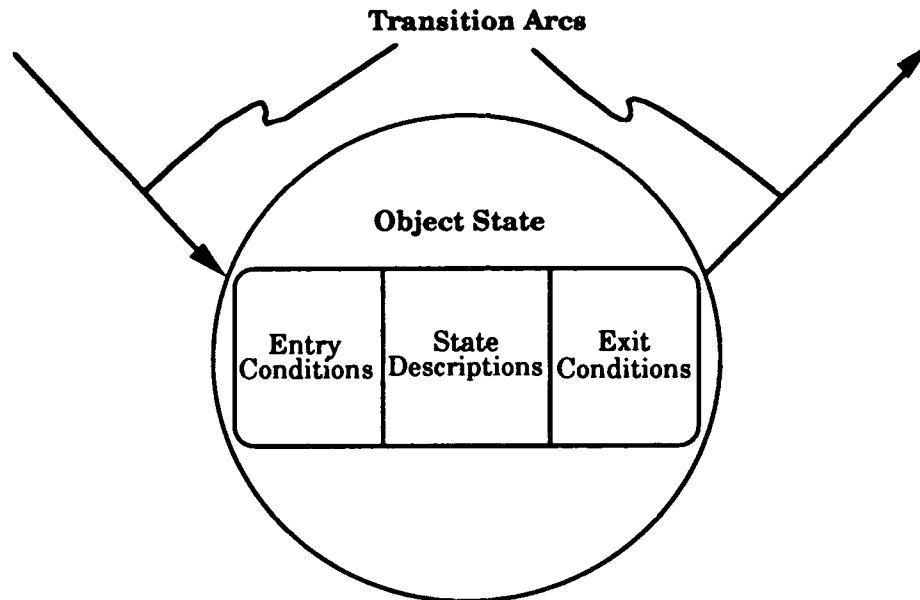


Figure 2-3

Object State Transition Network (OSTN) Diagram Concepts

As an example, consider the IDEF3 process flow diagram of the purchase order generation process for a fuel injection equipment manufacturing company shown in Figure 2-4. The *Request for Material* made by the production planning department initiates the material ordering process. If the requested material is an existing inventory item, an order for the required amount is placed on the current source of supply. If the material is new, activities to establish a new source of supply are initiated. This process consists of advertising for bids, receiving and evaluating the bids, and placing an order from the chosen supplier. The junction boxes containing an X (for eXclusive OR) indicate the choice of exactly one process flow path from several possible paths.

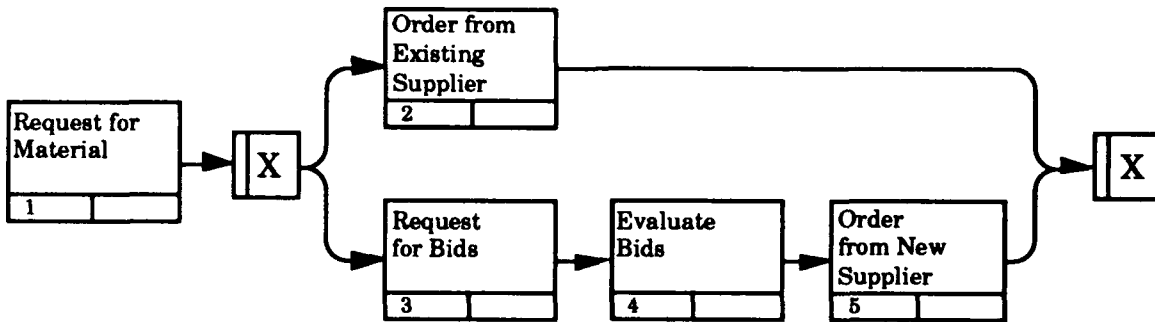


Figure 2-4
An Example Process Flow Description Diagram

A key documents in the purchase order generation process (see Figure 2-4) is the *Purchase Request Form*. This form is eventually transformed into a purchase order (PO) via the PO generation process. The OSTN diagram for this is shown in Figure 2-5.

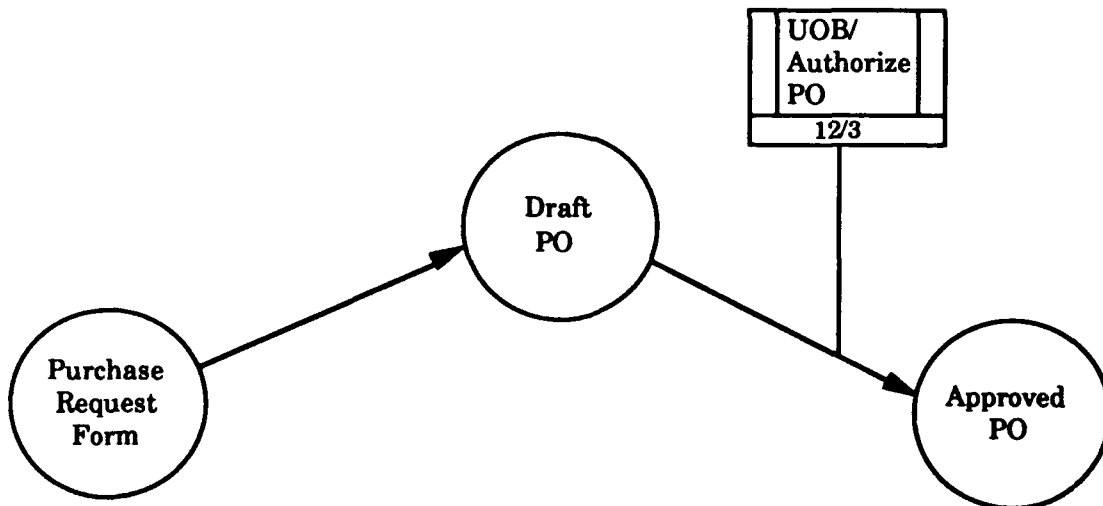


Figure 2-5
Object State Transition Network Diagram for the Purchase Request Form

Each circle in Figure 2-5 indicates a possible *state* that has been described for the object of focus. Associated with each state is an elaboration form called

the *Object State Description* (OSD) form which supports the capture of additional information such as how the object transitions to and from the state (the entry and exit conditions), as well as the defining features of the state. Thus, the OSD form for the state *Draft PO* would specify (among others) the conditions that would enable a transition from the *Purchase Request Form* state to the *Draft PO* state. The arrows that link the states in Figure 2-5 represent the *state transitions* from one state to another. The banded box labeled *Authorize PO* is an example of a *referent*. It is used to capture additional assertions concerning the transition conditions associated with the state transition arcs. In this example, the referent indicates that a Draft PO must go through an authorization procedure before it can be released as an Approved PO.

3.0 Basic Elements of IDEF3 Process Descriptions

The following sections describe the basic elements of the IDEF3 process description language. These elements, or building blocks, can be combined in many different ways to construct semantically rich descriptions of systems. An IDEF3 process description organizes the network of relations between actions in a specified scenario. Recall that IDEF3 descriptions are developed from two different approaches: process-centered and object-centered. Since these approaches are not mutually exclusive, IDEF3 provides a cross-referencing between them to provide a means of capturing and representing the totality of complex real-life process descriptions. Sections 3.1 through 3.5 contain descriptions of the syntactic elements of the IDEF3 process flow description language. Section 3.6 contains descriptions of the syntactic elements of the IDEF3 object state transition description language. The mechanisms for cross-referencing among statements made in each of these languages are introduced as part of the individual language specification. Examples interspersed throughout these sections illustrate how the basic syntactic elements are combined to build IDEF3 diagrams.

The basic syntactic elements of the IDEF3 process flow description language are shown in Figure 3-1. The basic building blocks of IDEF3 process flow descriptions are:

1. UOBs
2. Junctions
3. Links
4. Referents
5. Elaborations
6. Decompositions

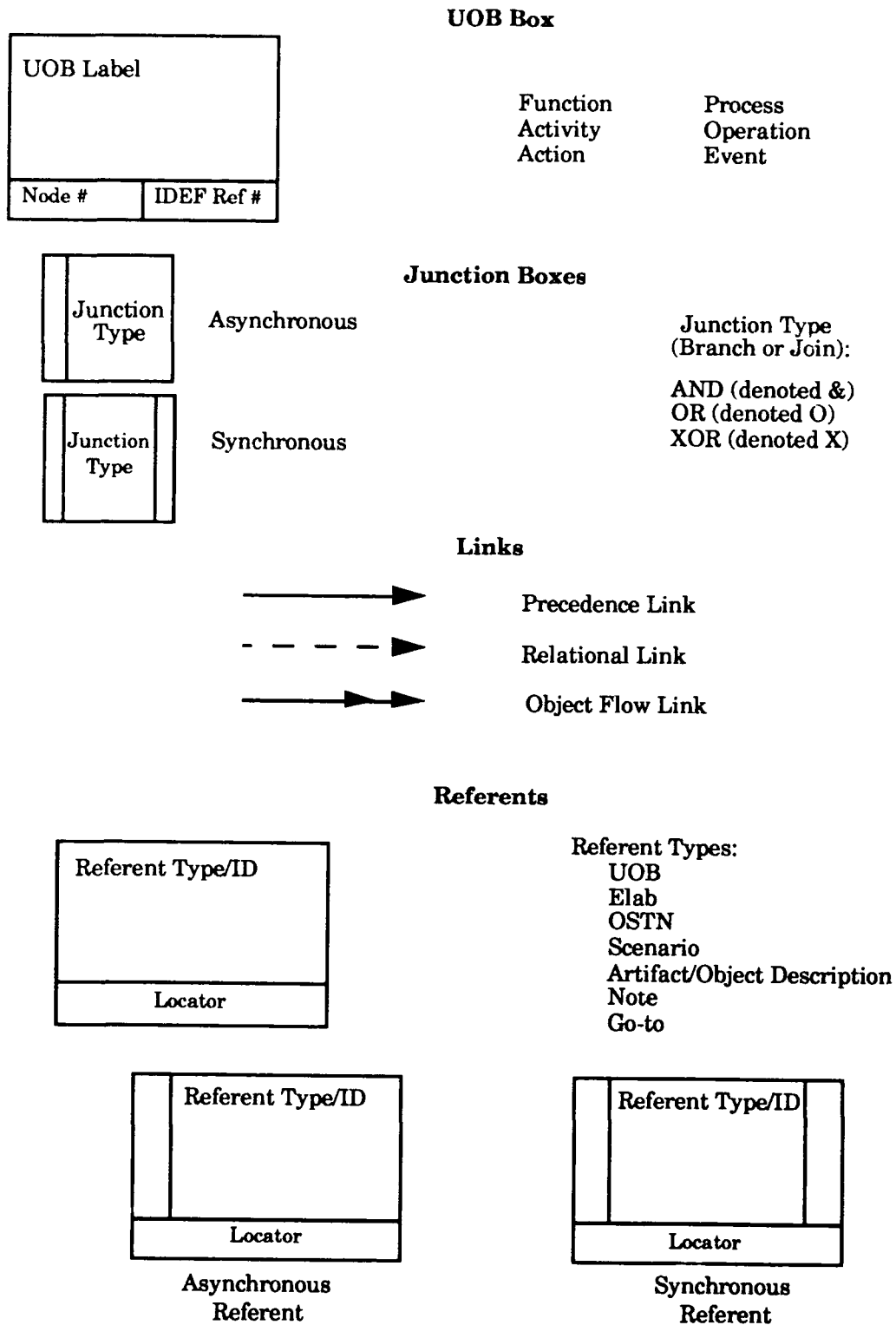


Figure 3-1
Symbols Used for IDEF3 Process Description Diagrams

An IDEF3 process flow description consists of a set of process flow diagrams and completed elaboration documents. Process flow diagrams contain statements constructed with the symbols that represent these basic building blocks. An IDEF3 process flow diagram is a representation of the assertions collected about the processing of a system expressed in a graphical language syntax.

A diagram displays a set of UOB boxes which represent activities, actions, processes, and operations of the real-world tied together with constraint links (arrows) to reflect precedence (solid arrow), user-defined relations (dashed arrow), or object flow (double-headed arrow). The logic of the process occurrence is captured through another type of symbol (the junction box) that can represent either the convergence (fan-in) or the divergence (fan-out) of multiple streams of process flow. Other supporting syntactic elements displayed in or associated with an IDEF3 diagram include: 1) boxes to indicate context dependent information (referents), 2) detailed specification forms for UOBs and links (elaboration forms), and 3) references to other diagrams (UOBs decompositions). In the following sections, each of these building blocks is described in greater detail and examples are provided to illustrate their use.

3.1 Units of Behavior

The capture of a description of "what's going on" within an organization or any complex system needs to account for a number of natural language concepts. Each of the following concepts is used in everyday language to describe "things that happen in the world."

1. Function
2. Process
3. Scenario
4. Activity
5. Operation

6. Decision
7. Action
8. Event
9. Procedure

Each of these concepts involves some circumscribed behavior. For instance, a reference to the *Planning Activity*, *Make or Buy Decision*, or the *Contract Award Event* carves up the world into spatio-temporal chunks to allow a description of "what is going on" in that chunk to be separated from the rest of the world. In IDEF3, a generic packet of information, or UOB encapsulates concepts such as those listed above.

In Figure 3-1, a UOB is represented by a special kind of box with a unique *label*. Each UOB can have associated with it: 1) a description in terms of a set of participating objects and their relations, and 2) descriptions in terms of other UOBs. The former is referred to as an *elaboration* of a UOB and the latter as a *decomposition* of a UOB. In the following two sections, each of these descriptive units will be outlined in more detail.

3.1.1 Unit of Behavior Elaborations

An IDEF3 process flow diagram graphically describes a process with the activities that occur in the process flow illustrated as boxes. However, a cursory inspection of these UOB boxes within a diagram will not provide a complete picture of the processes that are being described. Critical to the full understanding of a process flow description are the elaborations that are given for each of its UOBs. Elaborations provide the defining characterization of the real-world UOBs and are presented in the form of an *elaboration document*, (see Figure 3-2). The elaboration document identifies the *objects*, *facts*, and *constraints* that make up and control a UOB and provides for the inclusion of a textual description of the UOB. Every UOB has an elaboration document associated with it. In UOB descriptions, the elaboration document may consist of only a label and a reference number.

However, by adding more information to it, the elaboration document may provide the key to understanding UOBs that represent complex processes.

UOB Label	
UOB#	

Elaboration Document

UOB Name: _____

UOB Label: _____

UOB Number: _____

Objects: _____

Facts: _____

Constraints: _____

Description: _____

Figure 3-2
Unit of Behavior Elaboration Document

Figure 3-2 shows that the elaboration document comprises several fields, each representing different kinds of information. The following list contains a description of the contents of each of these fields.

1. **Document Identification:** This section consists of the Name, Label, and Number of the UOB being described. These

uniquely identify the UOB with which the elaboration document is associated.

2. **Objects:** This section lists the names of all the objects which participate in the process being described by the UOB. These objects can be either physical or conceptual. Objects can be created, modified, or destroyed during the course of the process. It may be useful to categorize an object as an agent, effected participant, or created or destroyed object.
 - A. **Agent** - if the object is the "do-er" of the UOB.
 - B. **Effected** - if the object is changed during the course of the UOB activity.
 - C. **Participant** - if no causality or transformation is associated with the object as a part of the UOB description.
 - D. **Created or Destroyed** - if the object is created or destroyed during the course of the UOB activity.
3. **Facts:** This section lists assertions about the UOB or the objects that participate in an occurrence of the UOB. Facts listed in an elaboration include characteristics of the objects (properties) and the relations that need to hold between objects during the course of the process. The fact list also includes properties of the UOB such as its duration, frequency of occurrence, or cost.
4. **Constraints:** This section contains a list of assertions about the limits within which a UOB operates. Constraints express the conditions that need to be met for an occurrence of a UOB to start, continue, or terminate. Constraints are groupings of facts or assertions that bound the UOB or govern the occurrence of an instance of the UOB. Constraints and facts are closely related. Constraints can often be distinguished from facts because they contain words indicating temporal or causal relations between activities. Examples of such words are *before*, *during*, *after*, *never*, *always*.
5. **Description:** This field contains a glossary entry (textual description) for the UOB. Typically, the glossary entry provides a textual recount of the information that is already in the object, fact, and constraint lists.

3.1.2 Unit of Behavior Decompositions

Elaborations capture and structure detailed knowledge about processes. If the process represented by the UOB is highly complex, it may be necessary to decompose the process into component (sub)processes. This “exploded” description, one level of less abstract detail, is called a *decomposition*. Decompositions are provided in IDEF3 to allow for capture of descriptions at varying levels of abstraction. Decompositions enable the application of the “divide and conquer” principle—a powerful mechanism for managing complexity. By applying this principle repeatedly, it is possible to structure the description to any level of detail required by the knowledge collected. Decomposition also provide the ability to model the same process from different knowledge sources or different points of views. This is possible because IDEF3 allows the same UOB to have a number of different decompositions, or “views.” This capability is useful in domain situations where a given process involves multiple functional organizations.

Syntactically, a decomposition is just another IDEF3 process flow diagram. Any or all of the IDEF3 building blocks can be used to construct a decomposition. In Figure 3-3, the use of decompositions is illustrated by an example drawn from the domain of processing contracts.

The decomposed UOB *Receive and Activate Contract* is called the *parent* UOB. Each decomposition of the parent UOB is a *child* decomposition. Moreover, each child decomposition is given a label and a unique number. The UOBs in a decomposition may also have decompositions.

Multiple view decompositions may be consolidated into an *objective* view. The view presented in Figure 3-4 is an example of an objective view of the UOB *Hold Kick-off Meeting*. This is the view perceived by a neutral observer of the Kick-off Meeting process. However, the project manager of the contract will have a different perspective of this process; therefore, IDEF3 enables him to express his viewpoint via an alternative decomposition of the UOB. The project manager’s decomposition of the UOB *Hold Kick-off Meeting* is shown in Figure 3-5.

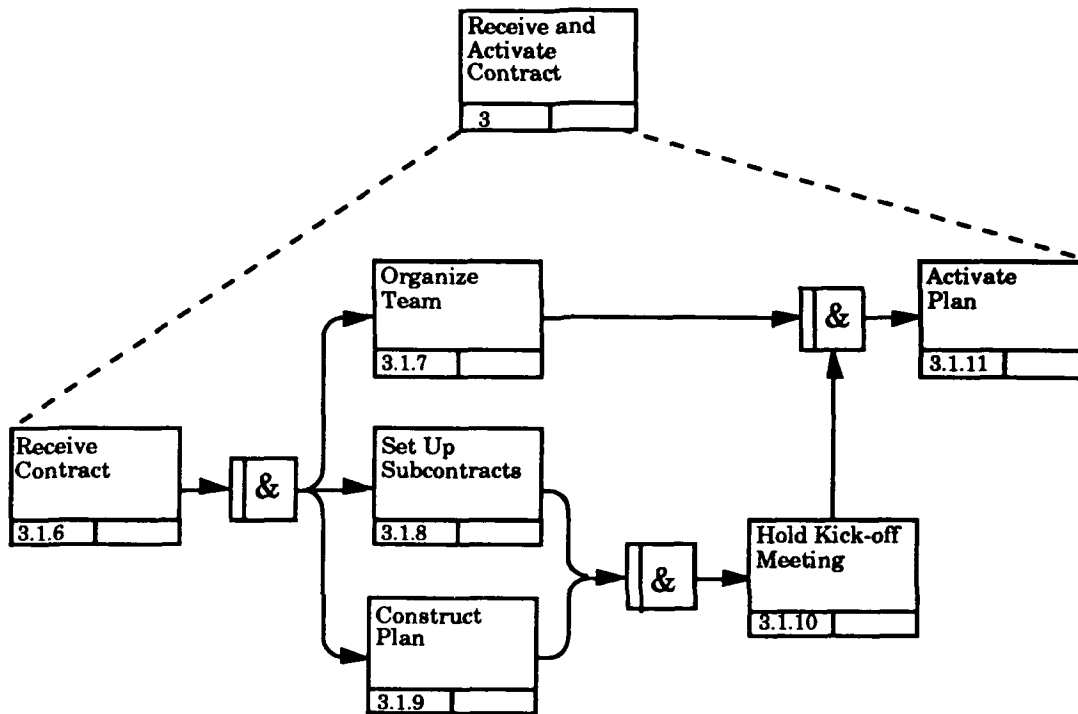


Figure 3-3
Decomposition 3.1 of Receive and Activate Contract

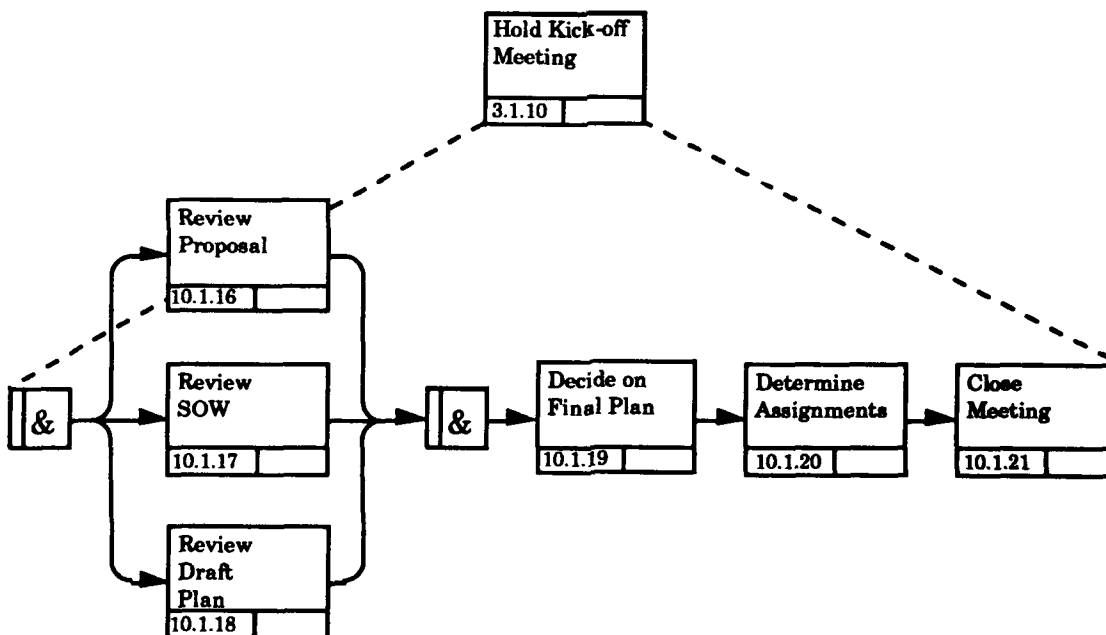


Figure 3-4
Decomposition 10.1 of Hold Kick-off Meeting UOB

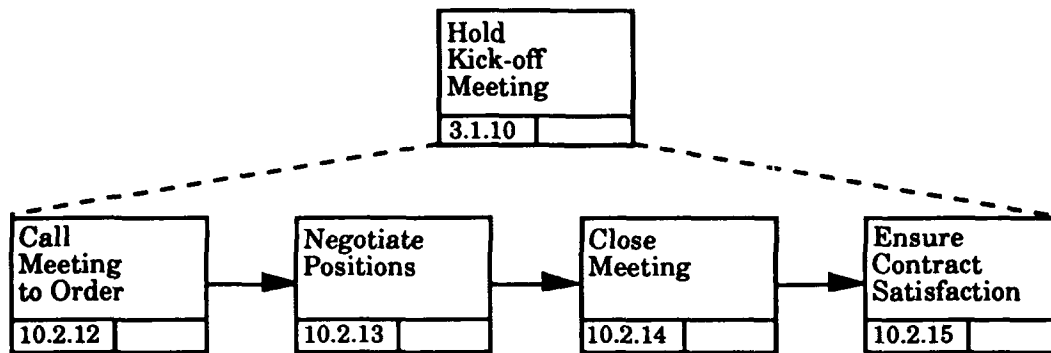


Figure 3-5
The Project Manager's View Decomposition

3.1.3 Unit of Behavior Numbering Scheme

A number is assigned to each UOB in an IDEF3 process flow description for reference and traceability purposes. With multiple decompositions and the large number of UOBs in a complex description, assigning a unique number to each UOB is imperative. Because of the complexities associated with referencing UOBs, a numbering scheme similar to that used for IDEF1 was adopted for IDEF3. During the development of the process flow description, UOBs are numbered sequentially in order of creation or discovery. Thus, within an IDEF3 process flow description (regardless of the number of scenarios), each UOB has a unique reference number.

It is also useful to be able to identify a UOB according to the context of its first occurrence (its parent). In a decomposition, each UOB reference number will have a prefix (formed from the parent UOB reference number) followed by a period, the number of the decomposition, and another period (see Figure 3-6). This numbering scheme enables each UOB in any decomposition to have 1) its own unique reference number within the total description, 2) a pointer to its parent UOB, and 3) an indicator of the parent decompositions to which it belongs. Note also, as illustrated in Figure 3-6, that UOBs do not have to be numbered sequentially from left to right.

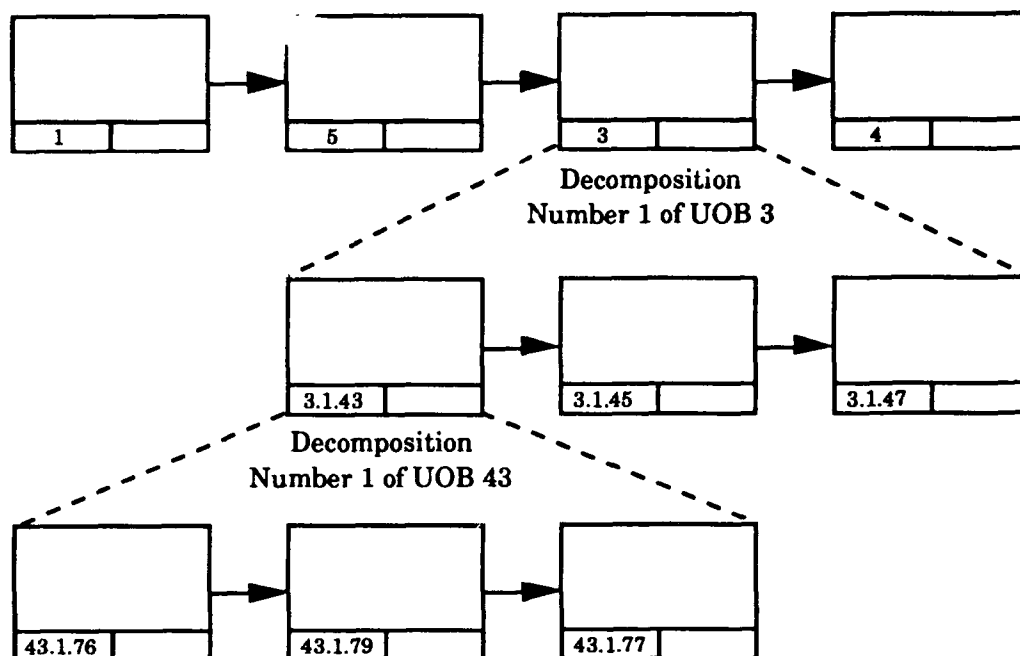


Figure 3-6
Unit of Behavior Numbering Scheme

If more than one individual is involved in creating the description, constraints are enforced on the assignment of numbers to ensure that every UOB is assigned a unique number. The procedure suggested for UOB number assignment is as follows. Each individual is assigned a set of numbers (e.g., Joe gets 1-99, Jane gets 100-199, etc.). Individuals can only assign UOB numbers from their allocated set. Once the initial set of numbers is used, additional numbers can be assigned as necessary. By enforcing this number assignment procedure, the lead analyst in the development effort can be assured that each UOB in the final combined description will contain a unique reference number.

3.1.4 Partial Descriptions

UOB boxes are joined together by links (see Section 3.2). Because of the description capture focus of IDEF3, it is possible to conceive of UOBs without links to other parts of an IDEF3 diagram, as the example in Figure 3-7 illustrates. These typically result early in the fact collection activity as

references are made by the domain expert to the existence of events or activities but no assertions have been made about how they fit together.

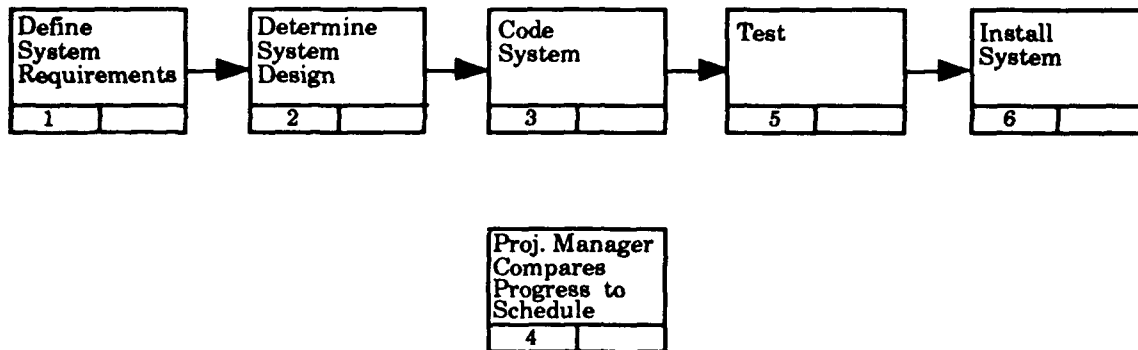


Figure 3-7
Disconnected UOB Example

In Figure 3-7, UOB 4 has no links to the rest of the diagram. This could either represent the actual situation or reflect the uncertainty of the domain expert's knowledge about the presence or absence of linkages. In this illustration, the diagram represents the actual situation rather than incomplete knowledge. The concept that makes the UOB *Project Manager Compares Progress to Schedule* part of this diagram is the object *Project Schedule* that is shared by other UOBs in the diagram. The IDEF3 method, by allowing the creation of such stand-alone UOBs, facilitates the creation of *partial descriptions*. It allows users to represent the state of the world as they know it, with no enforced constraints on completeness. In fact, a common error that can be committed in the course of developing descriptions is to attempt to "drive to completion" inherently incomplete knowledge sets.

3.2 Links

Links are the glue that connects the building blocks of the language. Links are used primarily to denote significant constraining relationships among UOBs. Links were added to the IDEF3 language to highlight constraints that are specified in the UOB elaboration. Links are intended to draw attention to important relations within an IDEF3 process flow description. The semantics

associated with the different kinds of links in IDEF3 allow for the representation of virtually any kind of relationship that could exist either within or between real-world processes. Examples of the types of relations that can be highlighted by IDEF3 links include temporal, logical, causal, natural, and conventional. The *link specification document*, enables the capture of additional details about a particular link. Links are drawn to start or terminate at any point on a UOB box or junction symbol (see Section 3.3). To enhance readability, process flow diagrams should be laid out so that links indicating the flow of objects (physical or information) or temporal precedence are drawn from left to right and top to bottom.

3.2.1 Link Types

The three types of links used in IDEF3 are *Relational*, *Precedence*, and *Object Flow*. The symbols that represent each type are shown in Figure 3-8.

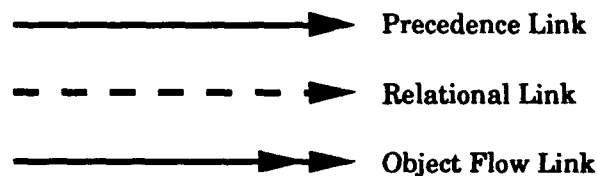


Figure 3-8
IDEF3 Link Types

Precedence links are a shorthand notation for expressing simple temporal precedence between the instance of one UOB and that of another. They are the most widely used link and are denoted by a solid arrow. When a precedence link connects two UOBs, the UOB instance at the start of the link completes before the UOB instance at the end of the link can start. Precedence links also imply an enablement relationship. If two UOBs are connected by a precedence link, an instance of the first enables an instance of the second.

Relational links carry no predefined semantics. For this reason, they are often referred to as *user-defined links*. This type of link merely highlights the

existence of a relationship between two or more UOBs. This relationship or constraint is specified in the link specification document described in Section 3.2.2. This type of link allows users to capture knowledge about a relationship without providing a structure to explicitly define that knowledge. The dashed arrow in Figure 3-9 indicates a user-defined relationship between the *Negotiate Changes* UOB and the *Accept Proposal* UOB. Although the negotiation of changes and the acceptance of the proposal occur in parallel paths, the interaction between these closely related activities is explicitly represented with the relational link.

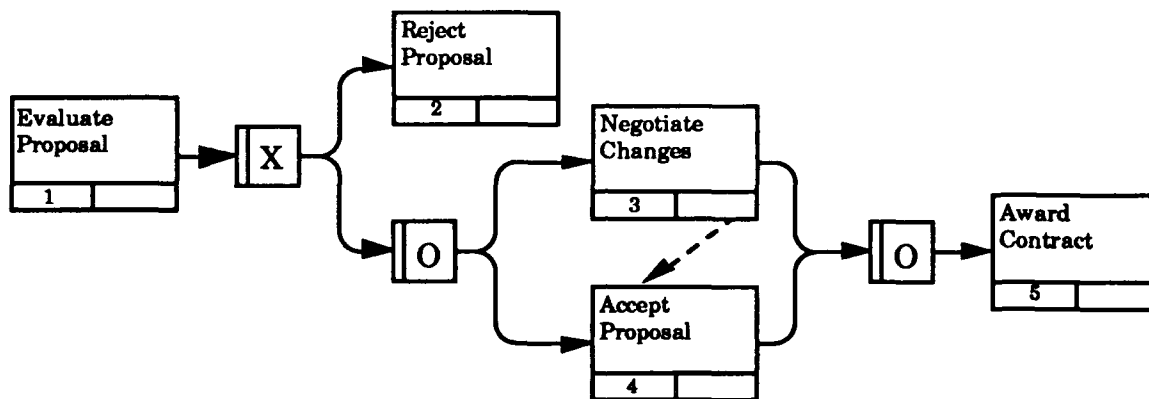


Figure 3-9
Example of a Relational Link

Object flow links provide a mechanism for highlighting the participation of an object in two UOB instances. This type of link carries the same temporal semantics as a precedence link. An object flow link is denoted by a solid arrow between a source UOB and a destination UOB, with a double arrow head point toward the destination UOB. It is important to note that the lack of an object flow link does not imply that the two UOBs do not share some object. The object flow link merely provides a means of highlighting a significant object flow relationship between two UOBs. An example application of an object flow link would be to emphasize that an object created in one UOB is critical to the completion of the process represented by another. A link specification document, as shown in Figure 3-9, will be

needed to clarify the reason for the object flow link and provide the name of the object that is associated with the link.

3.2.2 Link Specification Documents

Relational and object flow links are used to convey more information than simple temporal precedence between the participating UOBs. The special constraints on relational and object flow links are recorded in a link specification document (see Figure 3-10). This specification document is similar to a UOB elaboration both in format and purpose.

The following are descriptions of the sections of a Link Specification Document.

1. **Document Identification:** A link number that uniquely identifies the associated link and its link type. In addition the document identification contains a field for identifying the link type.
2. **Source(s):** Name(s) of the source(s) of the link. The source of a link is the IDEF3 box on which it starts. Multiple sources usually occur for links terminating at fan-in junctions. (See Section 3.3 for a description of junctions.)
3. **Destination(s):** Name(s) of the destination(s) of the link. The destination of a link is the IDEF3 box at which the link terminates. Multiple destinations usually occur for links originating from fan-out junctions.
4. **Object(s):** All significant objects that participate in the relationship that the link represents. These objects could include the objects within the source(s) and destination(s) of the link.
5. **Fact(s):** Significant characteristics of objects that participate in the relationship represented by the link. This includes both the properties of the objects relevant to the link, and the relationships known to hold between these objects.
6. **Constraint(s):** A characterization of the limits within which a link operates.

7. **Description:** The descriptive glossary associated with the link. Any descriptive information that does not logically fit into the other fields in the document is placed here.

Link Specification Document	
Link Number: _____	
Link Type: _____	
Source(s): _____ _____ _____	Destination(s): _____ _____ _____
Object(s): _____ _____ _____	_____ _____ _____
Fact(s): _____ _____ _____ _____ _____	
Constraint(s): _____ _____ _____ _____ _____	
Description (text): _____ _____ _____ _____ _____	

Figure 3-10
Link Specification Document

3.2.3 Link Numbers

Links that have link specifications need to be given link numbers. Links are numbered in a sequential manner. Prefixed to each link number is the letter *L* (for the word "link"). For example, the first numbered link is L1, the second is L2, and so on. The uniqueness of link numbers is ensured by using a procedure similar to the UOB numbering scheme. That is, link numbers are assigned sequentially from a pool allocated to an author. Link numbers are particularly useful in process flows with branches, for which it is convenient to describe the logic of branching in terms of the link numbers. Display of the link numbers on the process flow diagrams is optional.

3.3 Junctions

Junctions in IDEF3 provide a mechanism to specify the logic of process branching. Different junction types are provided in IDEF3 to aid in capturing the semantics of branching in real-world processes. Junctions support the description of 1) a process that splits into two or more process paths, or 2) two or more process paths will converge into a single process. Junctions simplify the capture of descriptions of sequencing and timing relationships between multiple process paths. Junctions do not provide the only means of capturing such descriptions. If the description cannot be represented in a clear or accurate manner using the predefined semantics of junction symbols, the analyst should use of specially defined relational links.

3.3.1 Junction Types

Junctions are classified in three different ways. First, they are classified according to the logical semantics conveyed: *AND* (&), *OR* (O), and *exclusive OR* (X). They are further classified as either *fan-in* or *fan-out*, based on whether they represent a *convergence* or a *divergence* in the logic of the process description. They are also classified based on the coordination of the timing of the associated UOBs as either *synchronous* or *asynchronous*. Figure 3-11 summarizes the relationships among the different classifications.

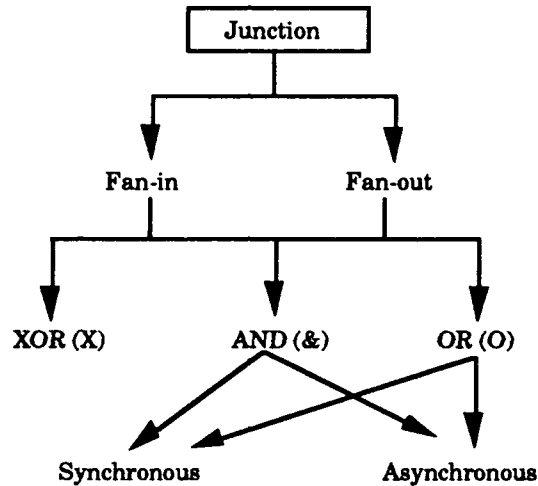


Figure 3-11
Anatomy of a Junction

3.3.1.1 AND, XOR, and OR Junctions

The classifications AND, OR, and XOR provide a standard logical interpretation to multiple processes through a junction. All UOBs leading to or from an AND junction will have to initiate or complete. An XOR junction indicates that exactly one of a set of possible UOBs will initiate or complete through a junction; an OR junction allows for some freedom of choice of alternative processes. The use of an OR junction implies that one or more of a set of UOBs will initiate (fan-out) or complete (fan-in) through a junction.

3.3.1.2 Fan-in and Fan-out Junctions

IDEF3 diagrams represent complex processes that often have multiple paths. Multiple process paths may initiate at a junction (fan-out) or terminate at a junction (fan-in). Fan-in junctions are junctions that represent the joining or converging of a set of different process paths. They are drawn with two or more links terminating at the junction. Fan-in junctions are classified based on the logic and timing of the terminating processes. The classifications of fan-in junctions are described in Figure 3-12.

Fan-out junctions represent the *splitting* or *diverging* of a process into a set of alternative processing paths. Fan-out junctions are drawn with several

precedence links leading out from them. The general semantics attached to a fan-out junction is that the set of outgoing processes would have to initiate in a manner that satisfies the constraints of the junction. The exact nature of these initiation constraints is determined from the type of fan-out junction. The different types of fan-out junctions and their associated semantics are summarized in Figure 3-13.






Fan-in Junction Type	Meaning
 - Asynchronous AND	All preceding processes must complete.
 - Synchronous AND	All preceding processes will complete simultaneously.
 - Asynchronous OR	One or more of the preceding processes will complete.
 - Synchronous OR	One or more of the preceding processes will complete simultaneously
 - XOR	Exactly one of the preceding processes will complete.

Figure 3-12
Fan-in Junctions and Their Semantics






Fan-out Junction Type	Meaning
 - Asynchronous AND	All following processes will start.
 - Synchronous AND	All following processes will start simultaneously.
 - Asynchronous OR	One or more of the following processes will start.
 - Synchronous OR	One or more of the following processes will start simultaneously.
 - XOR	Exactly one of the following processes will start.

Figure 3-13
Fan-out Junctions and Their Semantics

3.3.1.3 Synchronous and Asynchronous Junctions

The synchronization type of a junction specifies the relative timing of the process paths that either converge to or diverge from a junction. The interpretation is linked to the notion of an *activation* (or *instantiation*) of an IDEF3 diagram—a walkthrough of the process being represented which simulates its behavior as described on the IDEF3 diagram. For example, when a UOB is activated, it “starts,” “occurs,” and “completes.” After completion of a UOB, the process activation continues to the elements of the IDEF3 diagram after the UOB. Synchronous fan-in junctions indicate that the incoming processes must complete simultaneously (synchronously) before the UOB following the junction box can be activated. An asynchronous fan-in junction does not impose a timing constraint on the incoming process completions into the junction.

In summary, the semantics of a junction is dependent on whether the junction is 1) AND, OR, or XOR; 2) fan-in or fan-out; and 3) synchronous or asynchronous. In Section 3.3.1.4, the semantics of all possible combinations of these different categories is described in detail.

3.3.1.4 Junction Semantics

Key to the correct use and understanding of junctions is the recognition that the process flow diagram is a set of graphical language statements of collected assertions about *what happens in a process*. Figures 3-12 and 3-13 provide a summary of the semantics of the set of IDEF3 junction symbols. These semantics include elements of constraints employing logical operators, instantiation, and timing control. This section will further describe these different junction symbol semantics.

The semantics associated with the AND junction symbol are that all the processes leading out of a fan-out AND junction will eventually initiate. Similarly, for a fan-in AND junction, all the processes leading into the junction will terminate or complete. The two synchronization types are used to impose additional restrictions on the relative timing of the processes

activating through the junctions. All the processes leading out of a synchronous fan-out AND junction will have to initiate simultaneously. For an asynchronous fan-out junction, the processes leading out can initiate in any order. The conditions associated with a fan-in AND junction are only slightly different. All the processes leading to an asynchronous fan-in junction need to complete, but not necessarily with any particular order or timing. A synchronous fan-in AND junction requires the simultaneous completion of all the incoming processes.

The semantics associated with the asynchronous fan-out OR junction symbol is that one or more of the processes leading out of the junction will initiate in any order. For a synchronous fan-out OR junction, although any number of the processes can initiate at the junction, the processes that do initiate must do so simultaneously. The general semantics of a fan-in OR junction are that *at least one* of the preceding processes must complete before passing through the junction. For an asynchronous fan-in OR junction, the relative timing of the process terminations is unimportant; for a synchronous junction, the processes that complete must do so simultaneously.

The semantics associated with the XOR junction symbol are that exactly one of the processes leading out of a fan-out XOR junction will initiate. Note from Figures 3-12 and 3-13 that there is not a provision for specification of synchronous XOR. This is because an XOR junction provides for only one of the preceding or following UOBs to be instantiated, obviating the need for any type of synchronization.

3.3.2 Junction Combinations

The interpretations for the use of junctions in combination with each other are constructed from the base semantics of the types of junctions involved. In this section, a few typical examples of junction combinations that may occur in practice are presented.

Figure 3-14 illustrates one of the more frequently used junction types, the asynchronous AND junction. In this scenario, the completion of the receipt of

a proposal is followed by a cost and technical evaluation. In this process description, the technical and cost proposals must both be completed prior to contract award, but there is no specified timing relationship between the cost and technical evaluation. Both must follow the receipt and precede the award, but there are no timing constraints on either their initiation or termination.

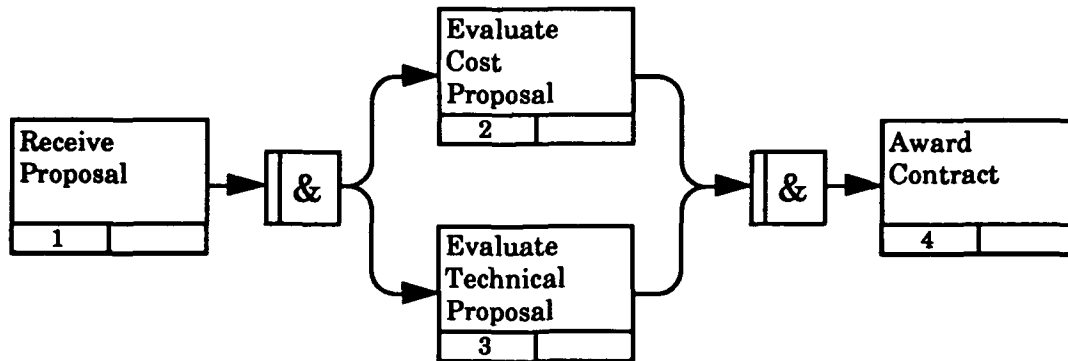


Figure 3-14
Asynchronous AND Junction Example

Contrast this to the scenario displayed in Figure 3-15 in which the synchronous AND describes a situation in which the cost and the technical evaluation *must* start simultaneously, but may end separately. However, if there had been an organizational rule that required both to end together as well, Figure 3-15 would have used a synchronous fan-in AND junction.

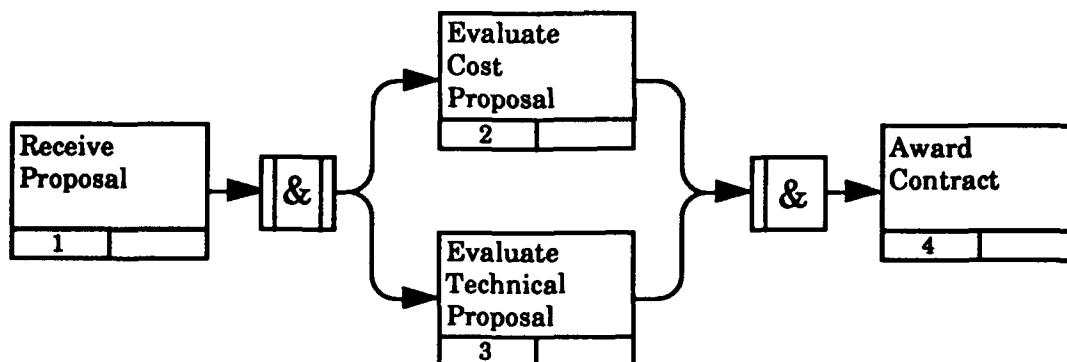


Figure 3-15
Synchronous AND Junction Example

Figure 3-16 shows a description of the proposal evaluation process. This process description states that, following evaluation, one can reject the proposal, negotiate changes, accept the proposal, or perform some combination of these.

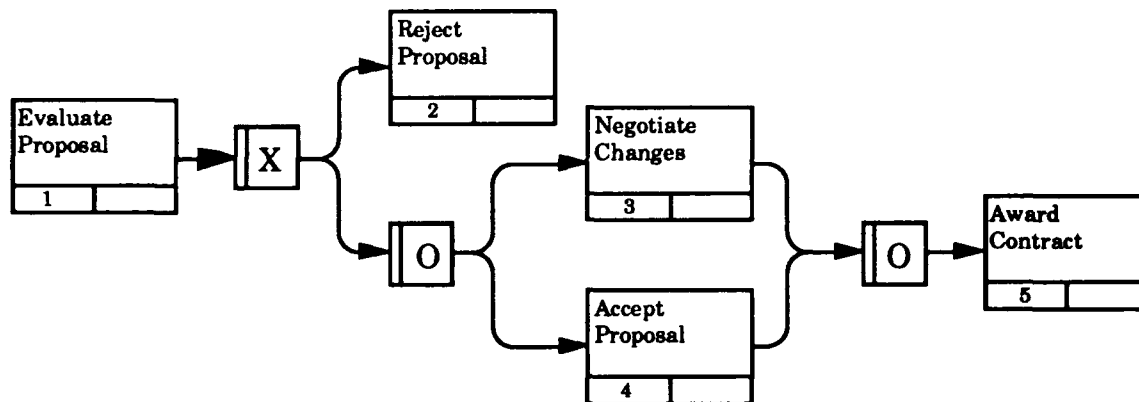


Figure 3-16
Asynchronous OR Junction Example

In the scenario depicted in Figure 3-16, *Reject Proposal* is a terminating activity; however, either of the other two activities (or both) will result in contract award. Note that a relational link indicates interactions between the *Negotiate Changes* and *Accept Proposal*. Note also that the above description is still partial in that it does not indicate what happens when the negotiations do not succeed. For example, in most situations, the award of the contract depends upon contractor acceptance of the terms of the funding agency. This may require the contractor to resubmit the proposal as a part of the negotiation process. Such additional information can be easily represented in IDEF3 as either additions to the current diagram or a decomposition of UOB 3.

3.3.3 Junction Numbering Scheme

To make unambiguous references to the junctions in an IDEF3 diagram, an identification scheme for IDEF3 junctions is provided. Recall that links are assigned unique numbers beginning with the letter L. Junction numbers

follow a identical numbering scheme, except that junction reference numbers start with the letter J. Thus, an IDEF3 process flow description may have junction numbers J1, J2, ..., Jn.

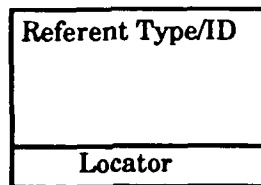
3.4 Referents

Referents allow the IDEF3 analyst to perform the following.

1. Span multiple pages or loop back in a diagram layout.
2. Refer to a previously defined UOB without duplication of its definition to indicate that another instance of a previously defined UOB occurs at a specific point in the process (without loop back).
3. Emphasize the participation of particular objects or relations in a UOB.
4. Tie in specific examples of referenced data or objects (e.g., screen layouts).
5. Associate special constraint sets to junctions. That is, associate an elaboration with a junction that contains additional facts, constraints, or decision logic which describe how that junction works.
6. Form references or links between the process flow diagrams and OSTN diagrams.

The graphical symbol of a referent is displayed in Figure 3-17. New IDEF3 users will often find referents an easy way to express ideas or concepts in lieu of junction types, dashed arrows, or constraint language statements.

The referent symbol syntax allows for three basic styles of referents as illustrated in Figure 3-18. The most commonly used style is the unconditional referent. An unconditional referent may be to a UOB, elaboration, junction, or object. Each type of referent may be used either in a process flow diagram or an OSTN diagram. Experience to date indicates that unconditional referents are most frequently used in process flow diagrams and the asynchronous and synchronous types are most frequently used in OSTN diagrams.



- Referent Types:**
- UOB - A Unit of Behavior on or off the diagram page.
 - Junction - A specific junction.
 - Object - An object of interest in the UOB to which the referent is connected.
 - Elab - An elaboration (normally used in the association of a referent with a junction.)
 - Scenario - A scenario an object must complete before changing states in an OSTN diagram.
 - Note - Additional user-specified information associated with the IDEF3 element to which the referent is connected.
 - OSTN - The object state transition network an object must complete before changing states in an Object State Transition diagram.
 - Go-to - An IDEF3 element to which processing will transfer (i.e., Go-to the UOB and continue processing from that point).

ID: UOB Label
 Junction Type (i.e., &, O, X)
 Blank (if it refers to an elaboration)
 Object Name
 OSTN Label
 Scenario Name

Locator: UOB#, Scenario#, Junction#, OSTN#, or Blank. For a locator of type UOB or Junction, the Locator should include either the Scenario# or the Decomposition# in which the ID occurs.

Figure 3-17
Referent Symbol Structure

The difference between the synchronous and asynchronous referents is based on the relative timing of the referenced element. The use of an asynchronous referent indicates that the referenced element needs only to initiate before the focus IDEF3 element (that is, the IDEF3 element that makes the reference) can progress to completion. The use of a synchronous referent indicates that the referenced element needs to both initiate and complete before the focus IDEF3 element can progress to completion. The following

paragraphs summarize the semantics of the possible forms of the referent symbol.

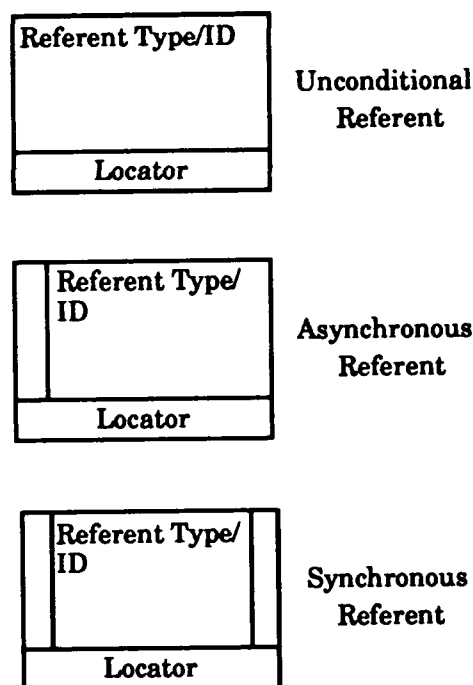


Figure 3-18
Referent Symbol Syntax

Unconditional Referents

If the referent type is "GO-TO" and the Identifier (ID) is a UOB Number, the next happening in the process flow is an occurrence of the referenced UOB. This type of referent is often used to document loops in a process flow.

If the referent type is "GO-TO" and the ID is a Junction Reference Number, the next happening in the process flow is an occurrence of the UOB(s) following the referenced junction.

If the referent type is "UOB," the ID must be a UOB Label; this means that another instance of a previously defined UOB occurs at a specific point in the process (without loop back). This type of referent is used to capture assertions processes that occur out of context.

If the referent type is "ELAB," the ID must be a Junction Reference Number, and this means there is an elaboration attached to the junction.

If the referent type is "OBJECT," the ID must be an object name; this means the named object participates in the UOB, Link or Junction to which the referent is attached. This referent type is often used to indicate the recipient of a transferred object as in the case of a design document being distributed to a number of different departments.

If the referent type is "OSTN," the ID must be an OSTN Label. If this is used within a process flow diagram, the completion of a UOB is conditioned on an object passing through some states of that OSTN. If this is used within an OSTN diagram this means that the state transition is conditioned on an object passing through some states of the referenced OSTN (see Section 3.6).

If the referent type is "SCENARIO," the ID must be a Scenario Name. If this is used within a process flow diagram, the next happening in the process flow is an occurrence of an activation of the referenced Scenario. If this is used within an OSTN diagram, the state transition is conditioned by an activation of the referenced Scenario (see Section 3.6).

If the referent type is "NOTE," the ID must be a user-constructed reference to a set of additional information that he wants to associate with the particular IDEF3 model element to which the note is attached. This referent type can be used to attach illustrations, text, screen layouts, comments, etc. to the description.

Asynchronous Referents

If the referent type is "UOB," the ID must be a UOB Label; this means that another instance of a previously defined UOB occurs at a specific point in the process (without loop back). If this is used within an OSTN diagram, an activation of the referenced UOB must be initiated before the state transition is allowed (see Section 3.6).

If the referent type is "OSTN," the ID must be an OSTN Label. If this is used within a process flow diagram, the completion of a UOB is conditioned on an object initiating transition through that OSTN. If this is used within an OSTN diagram, an object must initiate transition through the states of the referenced OSTN before the state transition is allowed (see Section 3.6).

If the referent type is "SCENARIO," the ID must be a Scenario Name. If this is used within a process flow diagram, the next happening in the process flow is an occurrence of an activation of the referenced Scenario. If this is used within an OSTN diagram, an activation of the referenced Scenario must start before the state transition is allowed (see Section 3.6).

Synchronous Referents:

If the referent type is "UOB," the ID must be a UOB Label; this means that another instance of a previously defined UOB occurs at a specific point in the process (without loop back). If this is used within an OSTN diagram, an activation of the referenced UOB must be initiated and completed before the state transition is allowed (see Section 3.6).

If the referent type is "OSTN," the ID must be an OSTN Label. If this is used within a process flow diagram, the completion of a UOB is conditioned on an object transitioning through that OSTN. If this is used within an OSTN diagram, an object must transition through the states of the referenced OSTN before the state transition is allowed (see Section 3.6).

If the referent type is "SCENARIO," the ID must be a Scenario Name. If this is used within a process flow diagram, the next happening in the process flow is an occurrence of an activation of the referenced Scenario. If this is used within an OSTN diagram, an activation of the referenced Scenario must complete before the state transition is allowed (see Section 3.6).

Note that referents can be used to avoid clutter on a diagram, provide additional data/information, or act as a note to the reader. The example in Figure 3-19 illustrates how a referent can be used to associate special constraint sets to junctions. This description states that, for certain

conditions, it will be required to loop back to UOB *PMAA* (*Perform Mission Area Analysis*). In this case, the referent on Junction J1 is used as a pointer to an elaboration that describes the conditions under which the referent *UOB/PMAA* would be activated.

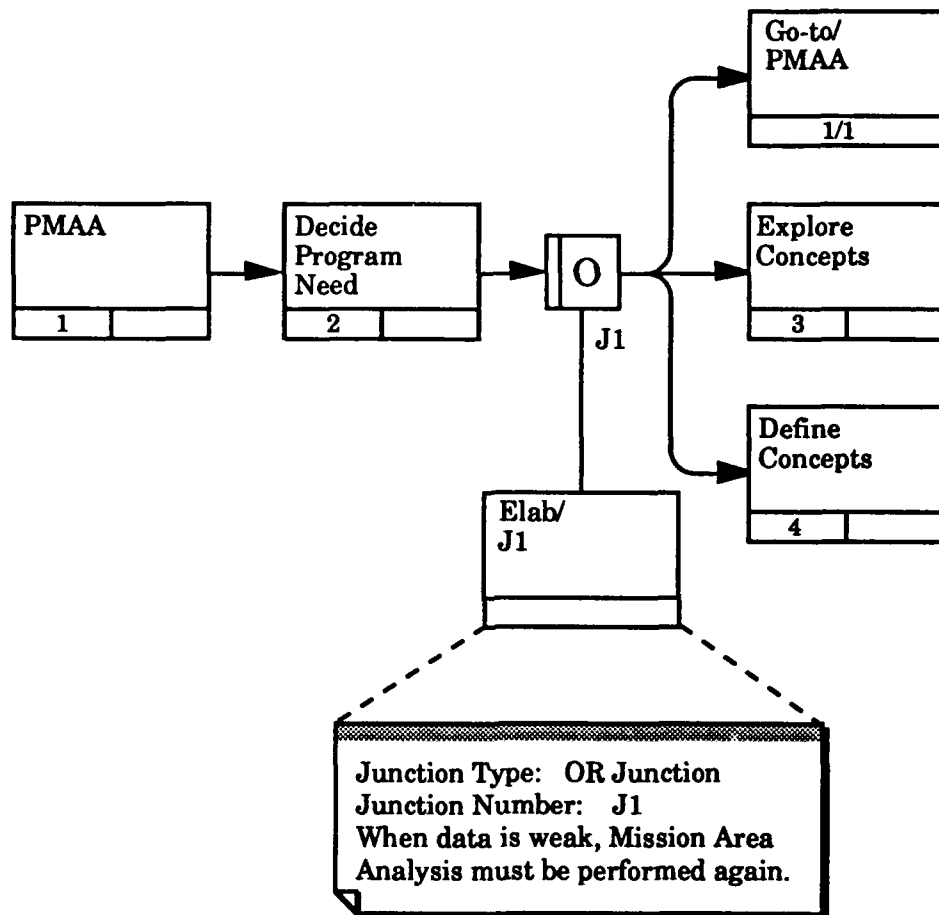


Figure 3-19
Junction Constraint Referent

A referent can be used to transfer control or indicate a loop back in the processing. The *Go-To/PMAA* referent loops the process back to the UOB *Perform Mission Area Analysis*. Finally, a referent can be used to refer to processing in a previously defined UOB without duplicating the definitions of the UOB (i.e., a referent of type UOB).

3.5 Combining the Basic Building Blocks

The syntactical elements of the IDEF3 graphical language are assembled in various arrangements to formulate the process flow description diagrams. As a guide for users, this section describes and illustrates some of the correct combinations of UOBs, links, and junctions that frequently occur in IDEF3 diagrams. A few examples are presented and explained to illustrate possible combinations of the IDEF3 building blocks.

The following definitions will prove helpful in fully understanding the examples in this section.

1. Activation (also called instantiation) is an occurrence of a process flow as described in an IDEF3 process flow diagram. Note that multiple activations of a process flow could occur in overlapping periods.
2. Instance is one occurrence of a process. Thus, an instance of a UOB means that the process represented by that UOB would occur one time.
3. Realization is the initiation followed by the completion of an occurrence of a process or an activation of a process flow.

To explain these terms further, consider the process flow *Respond to Customer Complaint*. An activation would be the initiation of the process of responding to a customer complaint continuing through all the UOBs given in the IDEF3 process flow description. An instance of *Record Customer Complaint* would be the one occurrence of this process within the overall process flow. *Record Customer Complaint* is realized when the recording is complete. *Respond to Customer Complaint* is realized when all the prescribed processes are completed for one customer.

3.5.1 Units of Behavior and Link Combinations

The following combinations of UOBs and links illustrate how links can be utilized to visually represent a relationship (e.g., temporal, precedence, etc.) between two UOBs.

Figure 3-20 illustrates a situation in which there is a well-defined precedence relationship between UOB 1 and UOB 2. The link in Figure 3-20 is a precedence link and implies that activity B will not begin until activity A terminates. This link does not preclude two activities A and B from acting on the same objects.

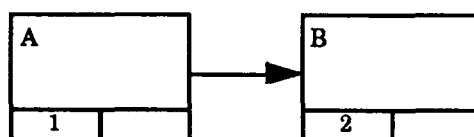


Figure 3-20
Precedence Link

In this form of UOB, link combinations can be used to:

1. Express simple temporal precedence between instances of one UOB type and those of another.
2. Capture that each instance of the predecessor UOB will complete before a paired instance of the successor UOB can begin.
3. Illustrate that in an activation of the process, if there is an instance of the predecessor UOB, there must be an associated instance of the successor UOB.

The dashed link in Figure 3-21 is a relational link. Since this link type carries no predefined semantics, the user must define the semantics according to the exact nature of the relationship between the participating IDEF3 elements.

Note that there are no predefined temporal semantics attached to the relational link, although the user can specify special temporal relationships using this link. The exact semantics of the relationship must be defined by the user in the link specification document attached to the link. For example, in Figure 3-21(a), the initiation and termination of the process indicated by the two UOBs may be illustrated by (b) or (c). Figure 3-21(d) illustrates a relationship between two UOBs that are not in the same process path. The link specification would contain the description of the special relationship.

Figure 3-22 illustrates the use of an object flow link. An object flow link has the same temporal semantics as a precedence link; however, its main purpose is to highlight the participation of an object in two different UOBs. It implies that the existence of such an object is of critical importance in the relationship between the two UOBs. For example, the object flow link in Figure 3-22 indicates that an object created by UOB A is essential to the performance of the activity represented by UOB B. A link specification would be used to explain the exact nature of the participation of the object in the two different UOBs.

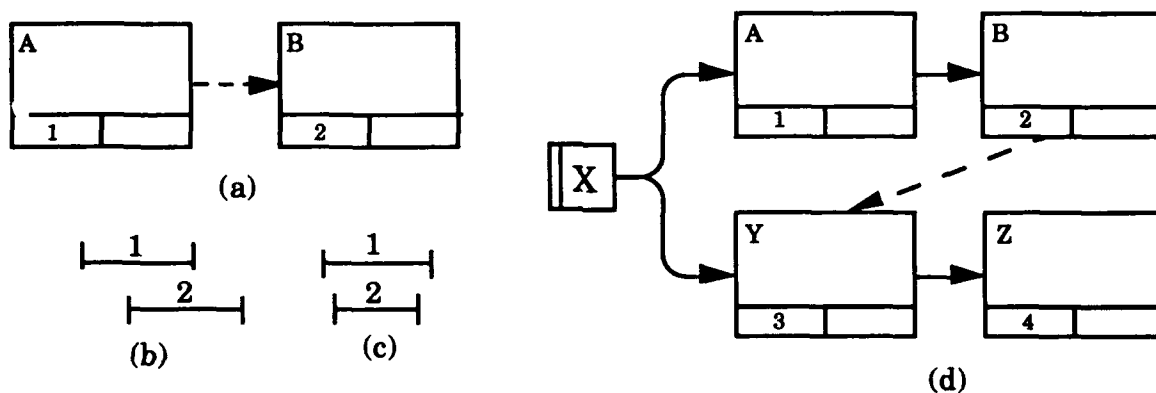


Figure 3-21
Dashed Link Example

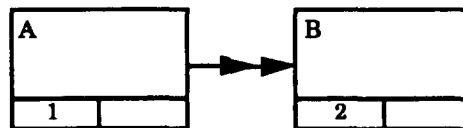


Figure 3-22
Object Flow Link

3.5.2 Combining Units of Behavior with Links and Junctions

The following combinations are representative of some of the most commonly found structures of UOBs, links, and junctions.

Link L1 in Figure 3-23 illustrates a precedence relationship between UOB 1 and junction J1. In the example, UOB 1 will complete before the decision logic of junction J1 will be realized (before the junction decision can be made). Junctions J1 and J2 are asynchronous AND junctions. J1 is a fan-out junction; J2 is a fan-in junction.

1. Fan-out (J1) Asynchronous: All three UOBs (2, 4, and 5) that follow J1 will start, although not necessarily at the same time.
2. Fan-in (J2) Asynchronous: Each preceding process (3, 4, and 5) must complete before the process activation can continue beyond J2.

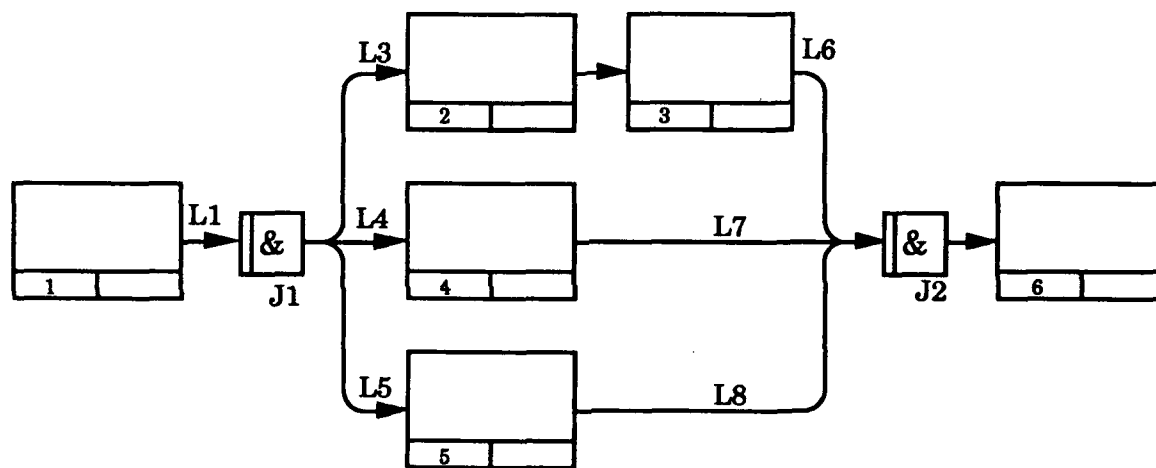


Figure 3-23
Asynchronous AND Junctions

In an activation of the diagram in Figure 3-23, the processing will proceed in the following manner. After the realization of junction J1, the three UOBs (2, 4, and 5) will activate--this activation can occur in any order. The asynchronous AND junction J2 will be realized only after UOBs 3, 4, and 5 complete. No order or timing of the completion is implied; however, the three UOBs must complete before J2 is realized. Finally, after the realization of J2, there will be only one realization of UOB 6 for one activation of the diagram.

The precedence link L1 shown in Figure 3-24 requires that UOB 1 be completed before the logic of junction J1 can be executed. In an activation, the synchronous AND junction J1 indicates that the processes represented by UOBs 2, 4, and 5 will initiate simultaneously. Likewise, the synchronous AND junction J2 indicates simultaneous completion of UOBs 3, 4, and 5 before processing continues past the junction.

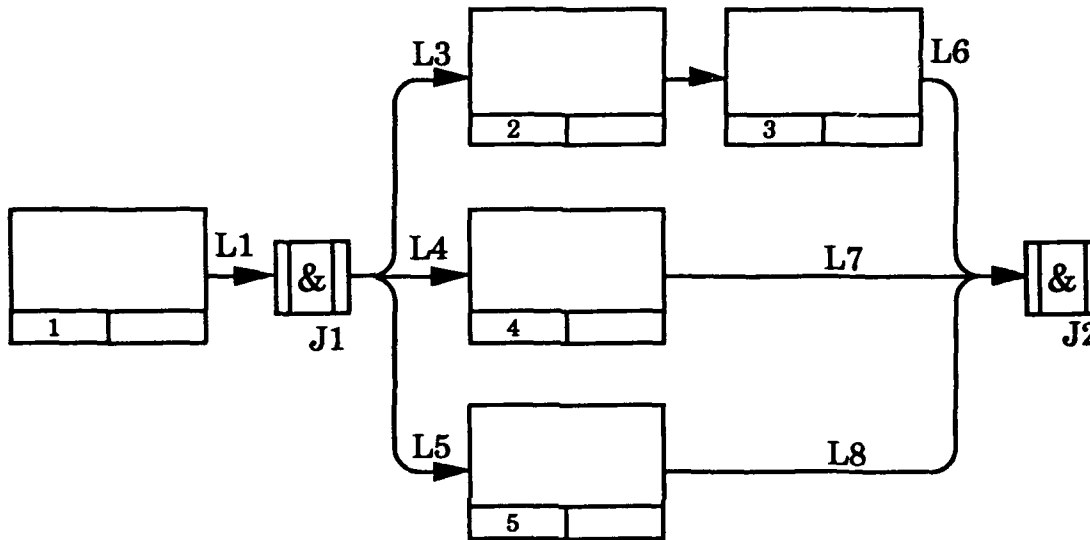


Figure 3-24
Synchronous AND Junctions

Figure 3-25 is structured like Figure 3-24 except that junctions J1 and J2 have become asynchronous OR junctions. In an activation of this process flow, the J1 OR junction indicates that one or more of the UOBs 2, 4, or 5 will be realized. This will initiate one to three process paths. The next junction to be considered in an activation is J2. Because J2 is an asynchronous OR junction, as soon as *one* of the paths completes, the UOB after the junction J2 *will* activate. This does not imply that in some activations more than one path will eventually complete before the realization of UOB 6. However, there will be only one realization of UOB 6; after its realization, any incomplete process paths in the structure will be ignored.

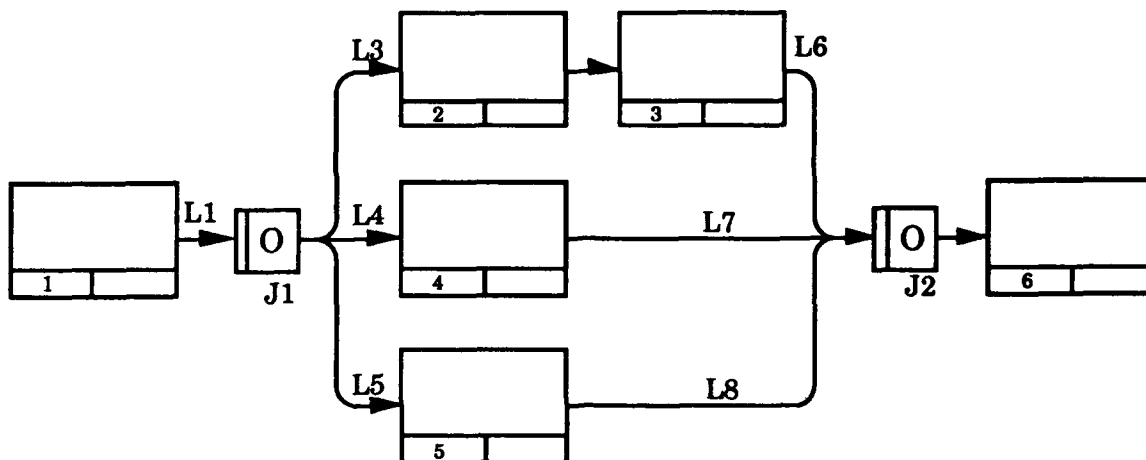


Figure 3-25
Asynchronous OR Junctions

Figure 3-26 illustrates the use of two synchronous OR junctions in combination. The fan-out OR junction implies that one or more of the UOBs 2, 4, and 5 will start. Since the junction is synchronous, when more than one UOBs is initiated, the initiations occur simultaneously. One or more of the UOBs 3, 4, and 5 will complete, and complete simultaneously--at the synchronous fan-in OR junction J2.

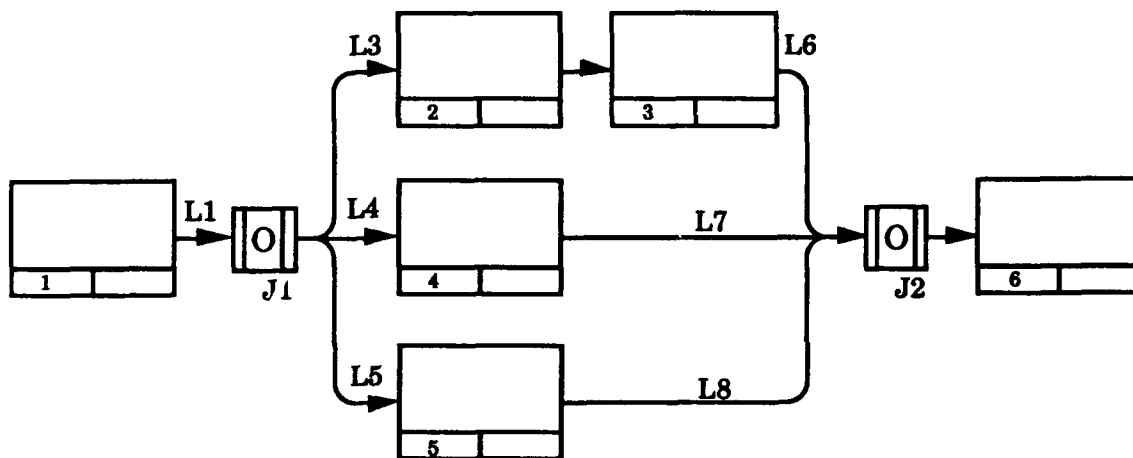


Figure 3-26
Synchronous OR Junctions

Note that for junction J2 to be realized and processing to continue through UOB 6, it is sufficient that *at least one* of the preceding processes (UOBs) complete. Consequently, the first UOB(s) to complete will successfully activate junction J2, while those UOBs (if any) that complete later, will be ignored. Consider the following sequence of events.

1. At time = t , UOBs 2, 4, and 5 are initiated.
2. At time = $t + 5$, UOBs 3 and 4 terminate at junction J2.
3. At time = $t + 10$, UOB 5 terminates at junction J2.

Junction J2 will be realized at time = $t + 5$, with the completion of the processes represented by UOBs 3 and 4. Since UOB 5 terminates later than UOBs 3 and 4, this UOB activation will be "lost."

Figure 3-27 is an example of a combination of two different types of junctions. Some of the valid UOB process completions for this example are as follows (the lower case letters represent instances of the corresponding UOB completions in Figure 3-27).

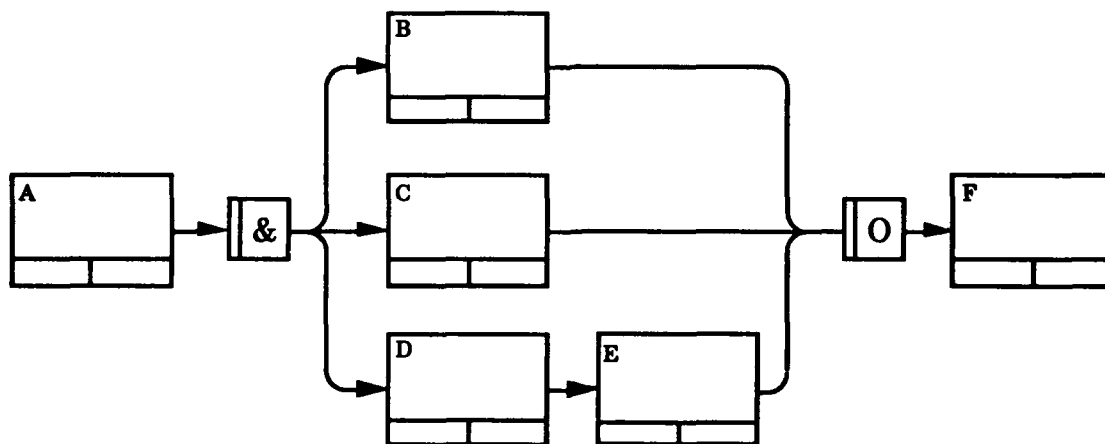


Figure 3-27
Fan-out AND Followed by a Fan-in OR Junction

1. (a, b, f)
2. (a, c, f)

3. (a, d, e, f)
4. (a, b, c, f)
5. (a, b, c, d, f)
6. (a, b, c, d, e, f)

Although UOBs B, C, and D succeed a fan-out AND junction, there are possible process sequences in which two or more of these UOBs may not complete before the process activates through the fan-in OR junction. This is because the fan-in OR junction is asynchronous. For a successful process activation through this junction, it is sufficient for the processing to have completed up to the fan-in junction from one of the paths leading into it. Thus, the completion sequence (a, d, e, f) indicates that UOBs B and C have *started but not completed* when E completed.

3.5.3 The Use of Referents in IDEF3 Diagrams

Referents may be unconditional, synchronous, or asynchronous. They enhance understanding, provide additional meaning, and simplify the construction of both process flow diagrams and OSTN diagrams. In this section, the use of referents in process flow diagrams is discussed; the use of referents in OSTN diagrams will be discussed in Section 3.6. The use of a referent to emphasize the participation of an object within a UOB is illustrated in Figure 3-28. In this example, the object *Milestone* is highlighted in the *Decide Program Need* UOB. This allows the diagram to graphically illustrate that the object *Milestone* is of importance in completing the activities indicated by the UOB *Decide Program Need*.

Figure 3-29(a) demonstrates a common use of referents to illustrate process logic out of fan-out junctions. After the OR junction, a Go-To referent is used to show the possibility of looping back to the *Perform Mission Area Analysis* UOB after completing the UOB *Decide Program Need*. This illustrates the use of a referent as a means to indicate a loop back in a process flow diagram. The junction referent in Figure 3-29(b) indicates that the processing after the UOB *Explore Concept* is transferred to the junction J4 in decomposition 2.1.

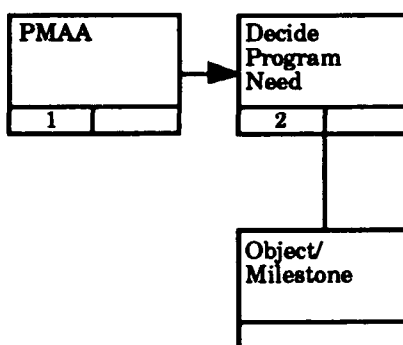


Figure 3-28
Object Referent Example

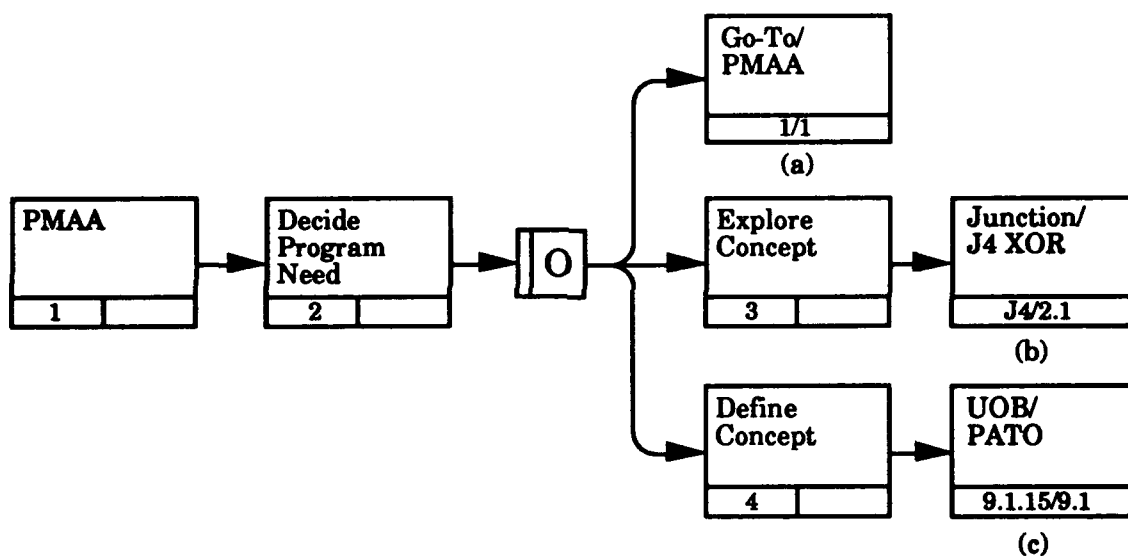


Figure 3-29
Referents in Process Flow Diagrams

Referents may be used to indicate that the process represented by a UOB in some other location is to be duplicated at some point. This use of a referent is illustrated in Figure 3-29(c). In the example, an instance of a process path that flows through the *Define Concept* UOB followed by the duplication of the processing that occurs in the UOB *PATO* (numbered 15) is found in decomposition 9.1. This duplication is indicated by the use of the UOB referent in Figure 3-29(c).

Referents are often used to highlight the participation of particular objects in a process, as shown in (c) and (d) of Figure 3-30. Figures 3-30(a) and (b) illustrate another important use of referents to display detailed specifications of junction logic. In (a) and (b), the logic of junctions J1 and J2 are modified using referents of type Elab(oration). Thus, elaborations are attached to junctions to express additional constraints associated with a junction.

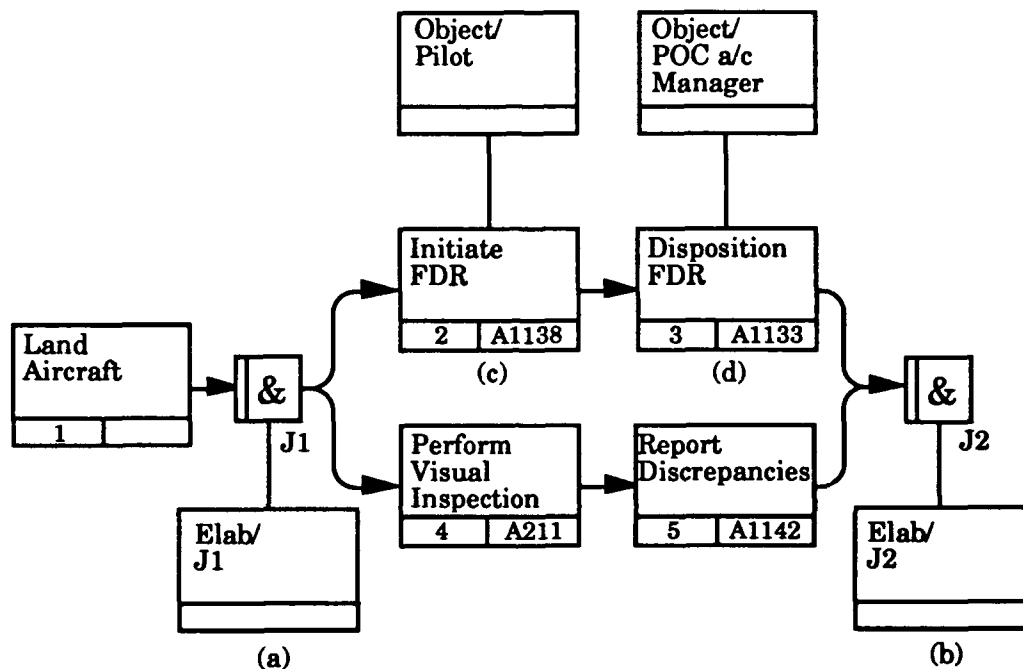


Figure 3-30
Referents to Elaborations

A more detailed description of the referent shown in Figure 3-30(a) is displayed in Figure 3-31. This elaboration reveals the logic of how the process flow diverges at fan-out junction J1.

3.6 Object State Transition Network Descriptions

OSTN diagrams are included in IDEF3 to allow for the capture of assertions concerning objects in the domain, states that those objects exist in, and conditions for state changes. This mechanism allows the construction of an object-centered view of a process. Object-centered views cut across the

process diagrams and summarizes the allowable transitions of objects in the domain. They have proven particularly useful in the documentation of data life-cycles.

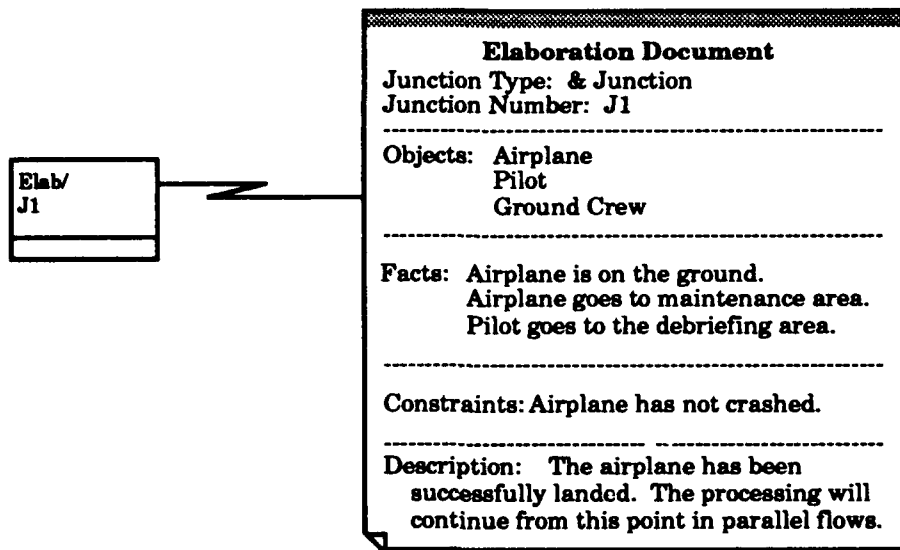


Figure 3-31
Referent to an Elaboration

Figure 3-32 shows the basic syntactic elements of an OSTN and Figure 3-33 illustrates the general form of an OSTN.

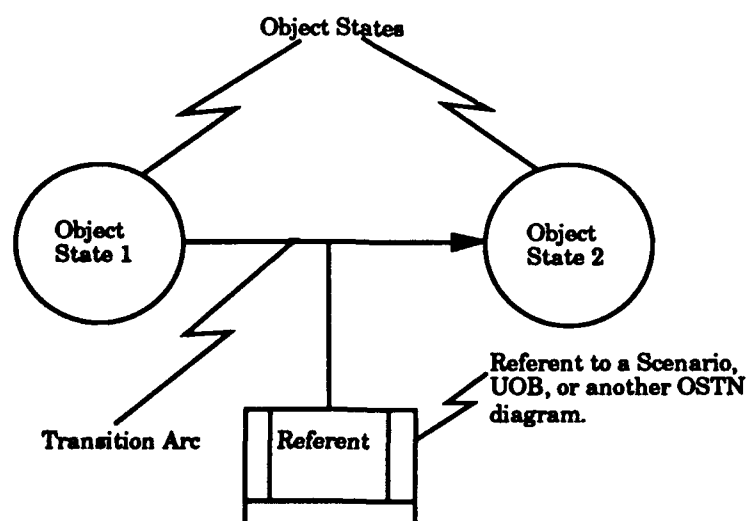


Figure 3-32
Symbols Used for Object State Transition Network Diagrams

Nodes (circles) represent object states. Arcs between the nodes represent possible transitions that the object can make between states. These transition arcs can have referents to scenarios, UOBs, or other OSTNs. The referents may be unconditional, synchronous, or asynchronous. Any object referenced in a UOB of an IDEF3 diagram can be characterized by an OSTN description. As a rule, OSTNs are created for only the most important objects. The semantics of each of these syntactic components and the OSTN grammar are described in this section.

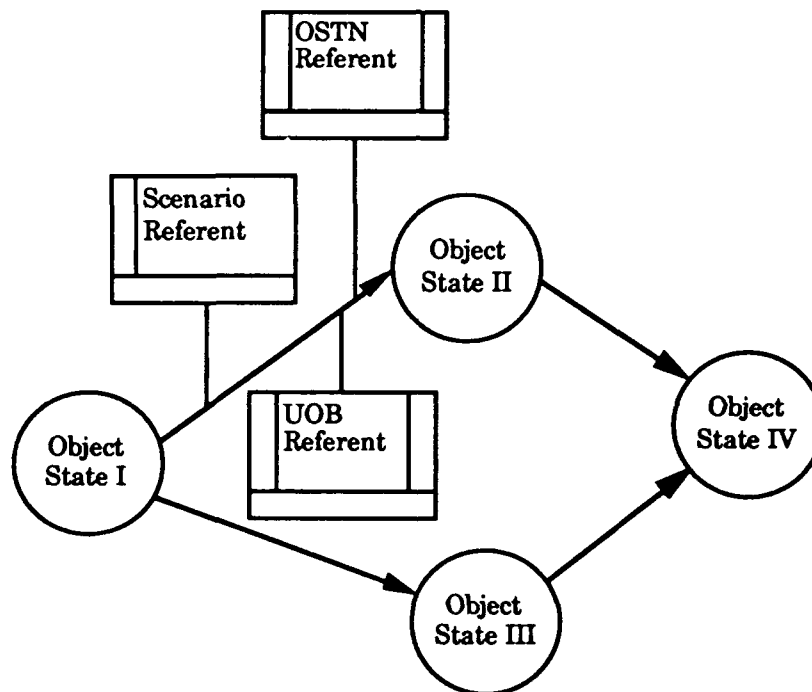


Figure 3-33
Object State Transition Diagram

3.6.1 Object vs. Object State

An OSTN diagrams focuses on objects and object states. An object can be 1) physical such as a report, part, or machine; 2) conceptual such as a decision, plan, or design concept; or 3) a combination of two or more physical or conceptual objects. Over the lifetime of an organization's interest in a particular object, the nature or characteristics of the object may change.

Thus, one object may exist in various states, each of which would be referred to as an object state. The state of an object is often identified by the individuals who work in the environment in which the object exists.

3.6.2 OSTN Description Components

The graphical representation of an OSTN description is an OSTN diagram (see Figure 3-33). Associated with each OSTN diagram is an elaboration specified by the OSTN Description Form. This form, shown in Figure 3-34, summarizes useful information about the entire OSTN network.

The following list contains a description of the fields that are contained in an OSTN description form.

1. **Object Name:** The name of the object that is the focus of this OSTN.
2. **OSTN #:** A unique identification number for the OSTN.
3. **OSTN Name:** The name of the object state transition network.
4. **Scenario Name:** The name of the scenario (if any) in which the OSTN resides.
5. **OSTN Label:** The string used to refer to this OSTN in an IDEF3 graphical element.
6. **OSTN Glossary:** A textual description of the OSTN, corresponding to the description field in the UOB elaboration documents.
7. **Object State Set:** The set of object states that make up this OSTN.
8. **Scenario Set:** Names of scenarios referenced in this OSTN.
9. **UOB Set:** Names of UOBs referenced in this OSTN.
10. **OSTN Set:** Names of other OSTNs referenced in this OSTN.

IDEF 3 Object State Transition Network Description Form	
Object Name: _____	OSTN #: _____
OSTN Name: _____	Scenario Name: _____
OSTN Label: _____	
OSTN Glossary: _____ _____ _____ _____	
Object State Set: _____ _____ _____ _____	OSTN Set: _____ _____ _____ _____
Scenario Set: _____ _____ _____ _____	UOB Set: _____ _____ _____ _____

Figure 3-34
Object State Transition Network Description Form

Another type of form, the Object State Description (OSD) form, is used to facilitate a detailed characterization of the object states that participate in an OSTN diagram. An OSD form is constructed for every object state represented in the OSTN diagram. In addition to enabling a detailed characterization of a state, the OSD form facilitates the specification of the requirements for all possible transitions in and out of the state as well as the requirements for the object to exist in a state. There are three types of requirements necessary to define a state: 1) *entry conditions*, those conditions that must hold for the object to transition into a state; 2) *state description conditions*, those conditions that need to hold while an object is in a state; and 3) *exit conditions*, those conditions that must hold for an object

transition out of a state. These conditions are expressed as attribute-value pairs and/or constraints. The OSD form is shown in Figure 3-35.

IDEF3 Object State Description Form	
Object State Name: _____	Object Name: _____
Object State Label: _____	OSTN Name: _____
Object State Set: _____ _____ _____	Glossary: _____ _____ _____
Entry Condition Sets: From State Name: _____ Facts: _____ Constraints: _____ From State Name: _____ Facts: _____ Constraints: _____	
State Description Conditions: <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> Facts: _____ _____ _____ </div> <div style="width: 45%;"> Constraints: _____ _____ _____ </div> </div>	
Exit Condition Sets: To State Name: _____ Facts: _____ Constraints: _____ To State Name: _____ Facts: _____ Constraints: _____	

Figure 3-35
Object State Description Form

The following list contains a description of the fields that appear on an object state description form:

1. Object State Name: Name of the object state.
2. Object Name: Name of the object.
3. Object State Label: String used in the object state symbol in an OSTN.

4. **OSTN Name:** Name of the OSTN that focuses on the object.
5. **Object State Set:** Object states that participate in transitions to and from this state.
6. **Glossary:** Textual description of the object state.
7. **Entry Condition Sets:** Conditions that must be true for an object to make a transition into this state. Must be given for *all transition arcs* leading to the state.
8. **State Description Conditions:** Conditions that need to hold to ensure that the object continues to reside in this state.
9. **Exit Condition Sets:** Conditions that must be true for an object to make a transition out of this state. Must be given for *all transition arcs* leading out of the state.

3.6.3 The Semantics and Use of Object State Transition Network Diagrams

As shown in Figure 3-32, the OSTN diagram consists of object states denoted by circles, transitions between these object states depicted by arcs (arrows) between the circles, and labeled referent boxes attached to those arcs. The network is thus a characterization of what happens to some object that is important in the IDEF3 description.

The OSD forms for each state provide a detailed specification of all the states in which the object can possibly be. These elaborations also specify the conditions for each transition that can occur to and from each state. The referents attached to the transition arcs provide an intuitive and graphical mechanism to specify the state transition conditions. There are three kinds of referents which can be attached to a transition arc: OSTN referents, scenario referents, and UOB referents. Each referent can be either synchronous or asynchronous. A referent attached to a transition arc means that the process implied by the referenced IDEF3 element must have been initiated prior to the transition continuing. If the referent is synchronous, the referenced process must start and complete before the transition can continue. An asynchronous referent implies that the referenced process must

start, but not necessarily finish, before the transition can continue. The order in which the referents occur along the transition arc specifies the order in which the referenced scenarios, UOBs, or OSTNs will occur. To illustrate, consider the OSTN diagram shown in Figure 3-36.

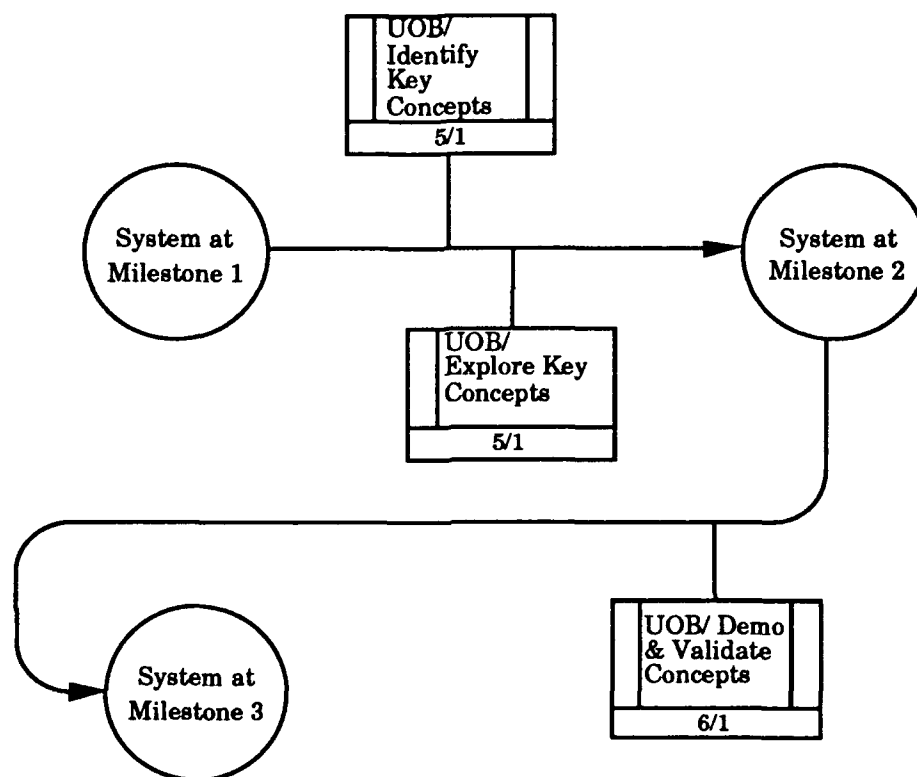


Figure 3-36
Example to Illustrate OSTN Concepts

The object of interest in the OSTN in Figure 3-36 is a system that can exist in three states: *System at Milestone 1*, *System at Milestone 2*, and *System at Milestone 3*. To transition from Milestone 1 to Milestone 2, the process implied by the UOB *Identify Key Concepts* must start and complete; the UOB *Explore Key Concepts* must then start. Similarly, the process called *Demo and Validate Concepts* must start and finish before the system can successfully transition from Milestone 2 to Milestone 3.

4.0 Development of IDEF3 Descriptions

This section presents a procedure for using IDEF3 as a process description capture, consolidation, and validation method. The procedure presented in this section is targeted at the needs of a large system description capture effort involving a team approach. Projects that are narrower in scope may not require all the activities described herein. The description development procedure is presented as a functional description. Experience with IDEF3 indicates that description capture is *similar* to knowledge acquisition and design endeavors. It is highly iterative, driven by findings, and often stylized by the participants. The activities described in this section should be considered "modes of thought" rather than sequential steps. The user should not expect to apply these activities in a strictly sequentially manner. With these disclaimers in mind, the framework presented in this section provides a default structure for first-time IDEF3 users.

The following roles are normally assumed by personnel involved in an IDEF3 process flow description capture process.

1. **Analyst:** The IDEF3 expert who will be the primary developer of the IDEF3 process flow description.
2. **Client:** The person or organization requesting the description development.
3. **Domain expert:** The knowledge source person in the application domain of interest.
4. **Primary contact:** The individual who acts as the interface between the analyst and the domain expert.
5. **Project leader:** The person ultimately responsible for the entire description development effort.
6. **Reviewers:** Persons knowledgeable in the domain and/or the IDEF3 method responsible for reviewing and approving draft descriptions and documents. Reviewers authorized to make written critiques of IDEF3 diagrams are *commentors*. The remainder are *readers*. Both team members and domain experts can be reviewers (see Section 4.5.1).

7. Team members: All personnel involved with the IDEF3 process flow description development project.

IDEF3 has been used to capture process descriptions across a wide range of domains including:

1. Gear production
2. Product design and engineering
3. Data life cycle management
4. Defense acquisition processes
5. Defense command, control, communication, and intelligence systems
6. Real-time control logic
7. Software man-machine interaction scenarios

The analyst often fixates on either the process description or the object state description and “gets stuck” particularly in domains (such as real-time control) in which a *process* is referred to by the domain experts as an *object*. IDEF3 is meant to provide a focused set of language mechanisms for organizing the fact statements acquired from a domain expert. The analyst should examine the entire set prior to initiating a project. Also, IDEF3 was designed to work with IDEFØ, IDEF1, and IDEF5. IDEFØ and IDEF1 are very useful in sorting out complex situations. IDEF5 provides additional description capabilities for recording ontology information (e.g., classification facts such as that a “carbide insert” is a perishable tool). In the course of a description recording effort, if the analyst begins to feel that the evolving description is awkward or misleading, he should step back and reevaluate the mechanisms being used.

4.1 Bounding the Description Capture Project

The development team must establish the purpose and context of the description capture effort as early as possible in the project. The context statement bounds or delimits the area of the domain addressed by the project.

The context is established by scope statements and the identification of the initial scenarios for the description capture project. The purpose statement provides a "completion criteria" for the description capture effort. The purpose is usually established by a list of 1) statements of objectives for the effort, 2) statements of needs that the description must satisfy, and 3) questions or findings that the client wants answered.

The purpose and context can rarely be determined completely and accurately in advance. The client often revises his list of needed findings or questions as the data starts being compiled. The area an analyst thinks will lead to the answer often turns up leads in other areas that were not considered within the scope. The purpose and context generally evolve during the initial part of the project. The purpose and context of an IDEF3 description are captured on an IDEF3 Description Summary Form shown in Figure 4-1.

IDEF3 Description Summary Form	
Project Title: _____ Project Leader: _____	
Purpose: _____ _____	Context: _____ _____
List of Scenarios: _____ _____ _____ _____	List of Objects: _____ _____ _____ _____

Figure 4-1
IDEF3 Description Summary Form

4.1.1 Defining the Purpose

Defining the purpose is an important initial step in the development effort. Often, project personnel take the purpose for granted only to find the results of their efforts ignored by or of little use to the client. Without a purpose statement, the only completion criteria is the budget and time allocated to the effort. Conversely, with a clearly defined purpose, the project can often be

completed in a budget much less than that anticipated. Defining the purpose can be separated into two parts, 1) defining a statement of need (SON), and 2) defining the information goals in terms of how that descriptive information will be used.

The SON should identify the source of the request (person or project) and paraphrase the stated objectives of the client. Identifying the information goals is simplified by answering the following questions:

1. Who will use the description once it is available?
2. What question(s) does the client need answered?
3. What issues are behind the need for the process description?
4. What decisions are behind the need for the process description?
5. How much detail is needed in the description to resolve an issue, make a decision, or answer a question?

4.1.2 Determine Initial Scope and Level of Detail

Once the purpose of the effort has been characterized, it is possible to define the context of the project in terms of 1) the scope of coverage, and 2) the level of detail for the description development effort. The statement of scope defines the boundaries of the description development effort. A project scope specifies which parts of the system are to be included and which are to be excluded. Ideally, the scope should select only those areas that are relevant to the needs of the client. An activity closely related to defining the scope is determining the level of detail of the description capture effort. The level of detail specification is normally documented in the form of a set of examples. It should be noted, however, that the scope and level of detail decisions are tentative at this stage of the project and should be updated as the description data becomes available. An astute project leader will regularly assess the adequacy of the description data captured against the specified needs and information goals of the client.

4.1.3 Identify Major Organizational Scenarios

In the application of IDEF3, another mechanism for establishing the context is identifying the important scenarios of operation within the scope of the project. For those familiar with IDEFØ, the scenario identification process is similar to the development of the A-0 diagram of an IDEFØ model. Identifying a scenario involves achieving a consensus among the team members on a title and paragraph description of a commonly occurring situation or problem that the system (organization) addresses. It is common for different scenarios identified to simply represent alternative viewpoints of (essentially) the same process. When possible, the beginning and ending UOBs of the scenarios should be established. Additionally, the activities that impact or feed the scenarios, but are outside the context of the description, should be identified to further refine the boundary of the description capture effort. While the statements of purpose and scope provide useful guidelines for successful completion of this activity, the insight of domain experts must be relied upon to actually identify the scenarios. The project leader should be aware that the scenarios identified are still at a very tentative level and some change can be expected as the data is collected and analyzed.

4.2 Collect Data

Once the context for the description development has been established, the stage is set for the actual data capture. The main information sources for data are the domain experts and source documents within the organization. The analyst must work closely with the domain experts to effectively capture data relevant to the description development effort. The data collection process is both iterative and interactive. Preliminary data provides guidelines for organizing the rest of the acquisition effort. The analysts interact with the domain experts and obtain descriptions, both written and recorded, of the process under study. The names of the activities and participating objects are extracted from these initial descriptions. Often, it is necessary to interview different experts who are knowledgeable about different aspects of the process. The data gathered from these interviews

must be carefully recorded so that the final description can be easily consolidated from these observations.

Other sources of data are often available, including IDEFØ function models, IDEF1 information models, data flow diagrams, and operational policy descriptions. The function models provide clues to UOBs and objects. IDEF1 models provide information that relates to the names of objects and the object states that are involved within the scenario. If related IDEFØ models are available, IDEF3 development time can be reduced by using the information collected either for the creation of the IDEFØ model or contained in the IDEFØ models. Activities of an IDEFØ model often correspond very closely to UOBs in an IDEF3 description. If IDEFØ models are not available, the IDEF3 developer can proceed without creating an IDEFØ model.

As data is collected during the course of the project, it should be logged on an IDEF3 Source Material Log as illustrated in Figure 4-2.

4.2.1 Identify Experts and Prepare for Data Collection

Together, the analyst and the primary contact identify a list of experts to be interviewed. No specific format for data collection is prescribed by the IDEF3 method. However, before the interview, the analyst should prepare a tentative agenda consisting of some specific questions. The following general guidelines are suggested to prepare for the interview.

1. Obtain background information about each expert from the primary contact. This includes information about the responsibilities, current assignments, and other areas within or related to the domain in which the expert has experience. The name, location, and telephone number of the expert(s) should also be recorded.
2. Prepare a brief outline of: 1) the purpose of the interview with the expert, 2) the topics to be covered, 3) the types of information being sought, 4) the authority for requesting the interview, and 5) the probing questions that can be used to motivate discussions.
3. Schedule a date and time for the interview with the expert.

USED AT:	ANALYST: I.M. Modeler PROJECT: Example IDEF3 Description NOTES: 12345678910 DATE: 28 Feb 93 REV:	WORKING	READER:		CONTEXT
			DRAFT	DATE:	
			RECOMMENDED		
			PUBLICATION		
Source Material #	Source Material Name/Description	Received From	Comments		
SM#1	Purchase Requisition/Form PI-R6 +-72	U.R. Buyer			
SM#2	Procedure #079-003 /Rev. 00 "Preparation of the Requisition"	U.R. Buyer			
SM#3	Procedure #079-001/ Rev. 00 "Preparation of the Purchase Order"	Policy and Procedures Manual			
SM#4	Procedure #101-506 "Purchasing Codes"	Policy and Procedures Manual			
SM#5	B.J. Commodity Code List	U.R. Buyer			
SM#6	B.J. Product Code List	U.R. Buyer			
SM					
SM					
SM					
SM					
SM					
TITLE: Source Material Log					NUMBER:

Figure 4-2
Source Material Log

4.2.2 Interview Domain Experts

The interview with the expert is critical. The analyst (interviewer) should create a positive, and friendly atmosphere during the interview. The interviewer should attempt to convey to the domain expert the feeling that they are working together to create the required description and solve some problem for the organization. A novice interviewer should constantly remind himself that the expert is the one with the knowledge of how a process should or does work. Generally, the expert is interested in helping and will often provide questions and lines of investigation that the interviewer had not thought of pursuing. The well-prepared interviewer will find that the expert will provide far more information than was expected, often covering topics the interviewer had not anticipated. In description capture, this is the bonus for good preparation. The expert often provides copies of documents and forms used in the process. This documentation may actually outline the process flow, or rather, the "Should-Be" process flow. The interviewer must remember that the main focus must be on the process actually performed, rather than formally documented procedures that are not followed.

Because IDEF3 is a process flow description capture method, the types of information that should be focused on during the interview include:

1. The constraints that govern the initiation of a process.
2. The conditions that must hold during the process.
3. The conditions that signal the termination of a process.
4. The processes that are triggered by the initiation or on termination of the process.
5. The properties of an occurrence of the process (e.g., duration, interruptability).
6. The objects that participate as agents, information, resources, or products in the process.
7. The properties of the objects (e.g., particularly those associated with the process such as arrival rates or spoilage rates).

8. Relations or associations between the objects in a single process.
9. The relations or constraints on objects between processes (e.g., shared resources).
10. The distinction between normal and exceptional situations in the occurrence of a process.

Collectively, the above set of information is referred to as "facts."

4.2.2.1 Collect Names of Objects

Under normal circumstances, one of the first types of information that an expert provides are the names of objects involved in the domain. The interviewer should carefully note all these objects. During the analysis that follows the interview, the analyst/interviewer will prepare a list of all these objects. This list, object pool, will later be analyzed further to associate the objects of the domain with the UOBs that are relevant in the domain.

4.2.2.2 Collect Names of Activities and Causality Relations

The named activities provided by the expert should be carefully noted. These will often become the names of UOBs that will be arranged to form the diagrams in the IDEF3 process flow description. As the names of the activities are collected, some notion of their sequencing and structure should be determined and noted. During the analysis that follows the interview, the analyst/interviewer will prepare a list of all these activities. This list is referred to as the "pool" of potential UOBs for the IDEF3 Process Flow diagrams. The analyst should also prepare a list of causality relations between the activities; this is the "pool" of constraining relations used to link the UOBs in the process flow diagrams.

4.2.2.3 Collect Situation Descriptions

In IDEF3, we refer to a situation description as the characterization of an occurrence of that activity. This characterization includes the association of activities with the collection of objects standing in particular relations during

an occurrence. It also includes the association of an activity with the other activities that precede or follow its occurrence. Situation descriptions can often be obtained by observing the process in action (e.g., visiting the factory where a particular part is made). However, such direct observation generally only provides information on the normal processing of short-duration situations. Generally, the analyst must rely on the domain expert to provide special insight, both into the normal processing of long-duration situations and the processing of exceptions to the norm. During the analysis of these situation descriptions, the analyst will add to the lists of objects and activities previously discovered. Analysis of the situation descriptions will provide the necessary insight into the sequencing of activities, the list of facts, and the constraints associated with the process to be described.

4.2.2.4 Maintain Description Findings Pools

During this phase of data collection and analysis, the source data is logged in the Source Material Log; the initial findings are cataloged into lists called pools. There are four pools used in IDEF3: 1) object pool, 2) scenario pool, 3) UOB Pool, and 4) object state pool. Figure 4-3 shows an example of an object pool. All other IDEF3 pools use the same basic layout as illustrated in Figure 4-3.

4.3 Formulate Process Flow Descriptions

At this point, the analyst should have lists of objects of interest, activities, facts, and constraints. Using this data and the situation descriptions, the analyst identifies the UOBs and begins to formulate the general structure of the IDEF3 diagrams. An initial process flow diagram is developed that illustrates the analyst's understanding of the information collected from the expert. Using the initial diagram, the analyst reviews the description with the domain expert to ensure the description is correct. These initial diagrams also assist the domain expert in recalling additional experience. The process of initial data collection is limited by the ability of the domain expert to recall his internalized knowledge.

USED AT:	ANALYST: I.M. Modeler		DATE: 28 Feb 93		X WORKING		READER:		CONTEXT:	
	PROJECT: Example IDEF3 Description				DRAFT					
	NOTES: 12345678910		REV:		RECOMMENDED		DATE:			
						PUBLICATION				

Object		Source	Object		Source
ID No.	Name	Material ID	ID No.	Name	Material ID
01	Purchase Requisition	SM1	018	Bill Of Material	SM38
02	Buyer	SM2	019	Route Sheet	SM40
03	Vendor	SM3	020	Destination	SM41
04	Purchase Order	SM15	021	Approver	SM43
05	Ship To Location	SM6			
06	Requester	SM11			
07	Department	SM12			
08	Pattern	SM21			
09	Part	SM26			
010	Purchase Req. Item	SM23			
E10	Commodity	SM30			
012	Purchase Req. Line	SM31			
013	Job	SM34			
014	Account	SM36			
015	Product	SM37			
016	B.M. Page	SM38			
017	B.M. Line	SM39			
TITLE: Object Pool					
					NUMBER:

Figure 4-3
Example IDEF3 Object Pool

Obtaining a description from an expert that is reasonably accurate and complete is an iterative process that must be repeated until the analyst's diagram agrees with the domain expert's knowledge. In some situations, it may be possible for the analyst and the domain expert to develop the descriptions together, rather than developing a draft description followed by a review procedure. The joint development approach can reduce the development time and produce descriptions that are more complete the first time, but the expert's time may be so valuable to the enterprise that he is rarely able to participate to the extent required.

The procedure for constructing a diagram (with or without active participation of the domain expert) is a six-step process.

1. Identify the UOBs.
2. Associate the UOBs with the appropriate scenario.
3. Organize the UOBs in a scenario into a basic causal sequence.
4. Add junction structures for logic description.
5. Develop elaborations for UOBs and link specifications as needed.
6. Develop decompositions for selected UOBs.

A key point to remember in constructing the process flow diagrams is that they are recording the facts that have been collected concerning the process. It is quite normal for them to initially not show a logical flow. A diagram often starts out with a set of UOB boxes with little connectivity among them. This is because the complete picture has not yet been acquired. It is not uncommon for the project to end successfully while there are still gaps in several of the diagrams. This can happen when the goals of the project did not require expenditure of the necessary information to fill those gaps. When using IDEF3 to capture descriptions, the analyst is not *designing* a system but rather organizing *known* facts about how a system works. Incredible as it may seem, there are many systems that work which have elements that no single person understands or even knows about.

4.3.1 Develop Unit of Behavior Elaborations and Link Specifications

The UOB elaborations and link specifications are developed from the interview data with review from the domain expert whose knowledge the description represents. Initially, the UOB elaborations and link specifications tend to look like simple glossary entries. However, as the data analysis progresses, they become similar to operation set-up sheets of a manufacturing process plan. All the facts that characterize the UOB concerning 1) participating objects and their roles, 2) relations, and 3) constraints that govern the UOB are described in the elaboration. These natural language elaborations will be written up on elaboration and link specification document forms. Information on these forms provides the most detailed characterization of the expert's description. The diagram is just a graphical presentation of a portion of this information.

4.3.2 Develop Decompositions for Selected Units of Behavior

A decomposition of a UCB is a collection of UOBs presented on a process flow diagram that provides additional or expanded details of a process represented by the parent UOB. In a base scenario process flow, the UOBs will usually have decompositions. The UOBs in these decompositions may also be decomposed. Different decompositions normally result from different domain expert views of what happens during an activity. They can also result from abstracting some participating object's view of the process. For example, a decomposition view might be created to show the processing steps required of the information system in order to support an organizational activity. Finally, decompositions can be produced by the analyst for selected UOBs to simplify a diagram. Thus, decompositions are diagrams providing a more detailed view or different perspective of a process, or a means to simplify a process description. It is important to note that the description development process is a refinement process. Decomposition development follows the same procedure as that for the primary description development. This refinement cycle consists of activities to 1) analyze the activity, 2) collect additional data,

3) describe situations in terms of related UOBs, 4) review, and 5) if necessary, return to a previous step in the procedure.

4.3.3 Cross-reference with IDEFØ Models

There is a definite relationship between the activities in an IDEFØ model and the UOBs in an IDEF3 process flow description. IDEF3 is not intended as a replacement for IDEFØ. If the system being analyzed is very large (e.g., Manufacture Aerospace Product), causal relations may not be evident. In these cases, it is often better to start with an IDEFØ model. Such a model can then be decomposed to a level where the precedence activities become prominent. On the other hand, if the facts collected do organize according to a cohesive story, it is generally better to formulate the IDEF3 process flow description first, then abstract an IDEFØ model from that description. The IDEF3 method was designed with this interaction in mind. The IDEF3 syntax recognizes this relationship by providing a means of referencing associated IDEFØ activities from within the IDEF3 UOB. As indicated in Section 3.1, all UOB boxes have a field (see lower right of Figure 4-4) for providing a reference to an activity in an IDEFØ model.

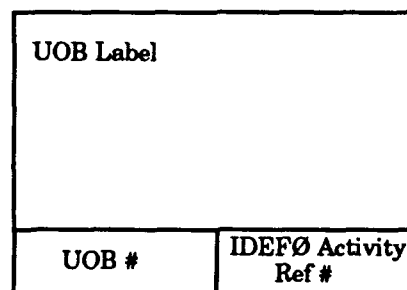


Figure 4-4
Unit of Behavior Fields

The reference scheme in IDEF3 assumes that zero, one, or many IDEFØ activities will map onto a single UOB. In cases where the UOB actually maps to only part of an IDEFØ activity, the activity referent should point to the set of child activities in the IDEFØ that are actually involved. If the IDEFØ is

not defined to a low enough level of detail, the extent of the mapping should be described in the UOB elaboration. As UOBs are identified, available IDEFØ references should be included.

4.4 Summarize Object State Transitions

OSTN diagrams are provided in IDEF3 to complement the process flow diagrams. OSTN diagrams enable an object-centered view of the system description by facilitating a detailed characterization of object states and state transitions. The development of OSTN diagrams may occur before, during, or after the development of the process flow diagrams. This section provides guidelines for the development of OSTN diagrams.

4.4.1 Select Objects of Interest

The first task in constructing the OSTN portion of a description is deciding which objects to describe. Basically, the analyst must identify which objects have state information and play an important role in the domain expert's knowledge about the system. The list of objects involved in a process may be extensive. In comparison, the list of objects of special interest is likely to be small. These will generally be objects that are modified by the process that is being described. Since the OSTN creation normally follows the process flow diagramming, a primary source for the objects of interest will be 1) UOB elaborations, 2) scenario descriptions, 3) IDEF1 or IDEF1X models of the information required by the scenario, and 4) original interview data. Regardless of the source of the objects, they have two features in common: 1) they undergo noticeable changes in the process, and 2) they exist in several states at various points in the process.

Since an object theoretically can be *any* physical or conceptual thing, there is no divining rod or scientific method for deciding which objects are in a domain. However, as a general heuristic, in IDEF3 we are interested in objects that play an important role in the operation of the system. Such objects will normally be named. That is, the analyst will find a word or

phrase that appears frequently in the interview information. Whatever this word or phrase refers to can be considered a possible object for consideration. The second issue to consider is whether the objects of interest have states of interest (obviously an OSTN diagram with no states would not be worth constructing). Again, some of the heuristics are: 1) each object state should display characteristics commonly recognized in the domain; 2) the object should be recognized to exist in a state for a period of time; and 3) there are recognized constraints or process that enable, cause, or inhibit the state changes. For each selected object, at least one OSTN diagram is developed.

4.4.2 Characterize Object State Transitions and Layout the Object State Transition Network Diagram

For each diagram, the creation of an OSTN description form is necessary. A textual description, or glossary, of the OSTN is part of this form. This text should contain a statement of the purpose for the diagram and will generally contain other information about the OSTN that does not readily fit into the other fields (e.g., ontology information that would later be included in an IDEF5 model). In addition to the textual description, the analyst records the object states and the other IDEF3 elements (UOBs, process flows, and OSTNs) that are referenced in the diagram. Initial completion of this form is part of the analysis activity associated with construction of the OSTN diagram. Although the form will not be completed at this time, this initial work aids the analyst in developing an OSTN diagram from the raw data.

The next step in OSTN diagram development is to describe each object state, and characterize the state transitions. To accomplish this, the analyst will perform the following tasks.

1. Identify the defining characteristics for each object state.
2. Identify the criteria for entering each state.
3. Identify the conditions for leaving each state.
4. Identify the possible transitions between states.

5. Identify special conditions for enabling each state transition.
6. Identify the activities that cause, allow, or are caused by each transition.

The results of the first three activities are recorded on the object state description (OSD) form for each affected state. The results of the last three activities determine the network layout. Once this analysis is complete, the OSTN diagram is created; the OSD forms and OSTN descriptions are then modified.

4.4.3 Cross-reference to IDEF1 Models

In many large IDEF3 development projects, IDEF1 and/or IDEF1X models are available prior to the project initiation. If these are available, they can help identify the objects for which the OSTN diagrams are drawn. The IDEF1 model and entity class number or attribute class that relates to each object or object state should be referenced in the glossary of either the OSTN or the appropriate OSD form.

4.5 Validate IDEF3 Process Descriptions

4.5.1 Motivation

The leverage of IDEF3 for description capture is revealed when the validation² stage is reached. Conventional process modeling techniques tend to discourage the capture of incomplete or inconsistent system descriptions through the use of rigid syntactic or semantic mechanisms. Furthermore, they force the user to gloss over gaps in the description or simplify the facts with idealizations. IDEF3 does not impose such restrictions. It provides a

² The genesis of this kit review procedure comes directly from the original IDEFØ Function Modeling "yellow book," AFWAL-TR-81-4023. This was done to maintain consistency among the IDEF methods. The input from this document is greatly appreciated and acknowledged.

flexible yet formal mechanism for recording the facts known about the operation of the system. Gaps and inconsistencies are made obvious in the diagram layouts specifically to bring them to the attention of the analyst and the domain experts. Likewise, capture and display of multiple viewpoints of the process documents these differences and generally leads to discussions/negotiations between the domain experts to resolve differences or decide on a harmonization. A better understanding of the process is achieved by both the experts and the analyst as they attempt to complete gaps and resolve inconsistencies both in a view and between views. This creates an understanding of how perceptions about the process differ between experts. In contrast, conventional techniques typically present the analyst's assumptions about the process interspersed with his understanding of the expert's description. This model is then presented in a voluminous and unreadable format for validation. Often, the expert, either in the interest of expediency or because of increasing pressure for a consensus, signs off on a system process model without completely understanding the implications. Using IDEF3, it is possible to use the system description diagrams as discussion focal points to resolve inconsistencies (if any) between the user's and analyst's differing viewpoints of how a process works.

4.5.2 Build and Distribute Kits

A primary means of validating IDEF3 process descriptions is through the review and approval of kits. Kits represent portions of the total description that have reached some state of completion. The kit review task can be performed anytime during the description development effort as a mechanism for acquiring additional facts or when a significant portion of analysis work has been completed (e.g., completion of the initial lists of UOBs and objects, completion of one or more OSTN diagrams, completion of a process flow diagram). Kit production and the associated review cycle discussed in Section 4.5.1.2 provide a disciplined approach that will result in an accurate description of the process and subsequently produce a final product that will satisfy the goals of the project.

4.5.2.1 Roles in the Kit Review Process

The roles described earlier in Section 4.1 are further specialized for the kit review process. The roles of the personnel involved in the kit review process are as follows.

1. **Analyst:** IDEF3 expert who is the primary developer of the IDEF3 description. The review process initiates and terminates with the analyst. The analyst relies on the domain expert for the technical content of the description, during both description capture and the kit review cycle.
2. **Reviewers:** All personnel involved in the review of IDEF3 kits.
3. **Commentors:** Reviewers who are not only knowledgeable in the application domain but also proficient enough in IDEF3 to offer structured comments in writing. Commentors read the material produced by analysts and verify its technical accuracy. They are domain experts and are responsible for finding errors and suggesting improvements in the IDEF3 process flow description. The role of a commentor is key to producing high quality results. The commentor determines whether the purpose has been adhered to, and whether errors or oversights exist. Commentors are authorized to make written suggestions during the review process.
4. **Readers:** Reviewers to whom IDEF3 kits are distributed for informational purposes only. Readers are often individuals from whom analysts may have obtained information via interviews.
5. **Librarian:** A person assigned the responsibility of maintaining a file of documents, making copies, distributing IDEF3 kits, and keeping records.

A "role" is not related to an individual's job title; therefore, the same person may be asked to perform several roles. Thus, each individual's participation is, in fact, unique and depends upon the IDEF3 kit involved.

4.5.2.2 The IDEF3 Kit Review Cycle

Kits represent portions of an IDEF3 process flow description that have reached some state of completion. These draft portions of a description are

distributed for review in the form of a standard IDEF3 kit. The IDEF3 kit review cycle illustrated in Figure 4-5 is based on the kit review process for other IDEF methods. For clarity, the following steps do not mention the librarian, but focus on the interaction between the analyst and commentor. With large systems, the role of the librarian is essential. In smaller efforts, that role may be assumed by the analyst.

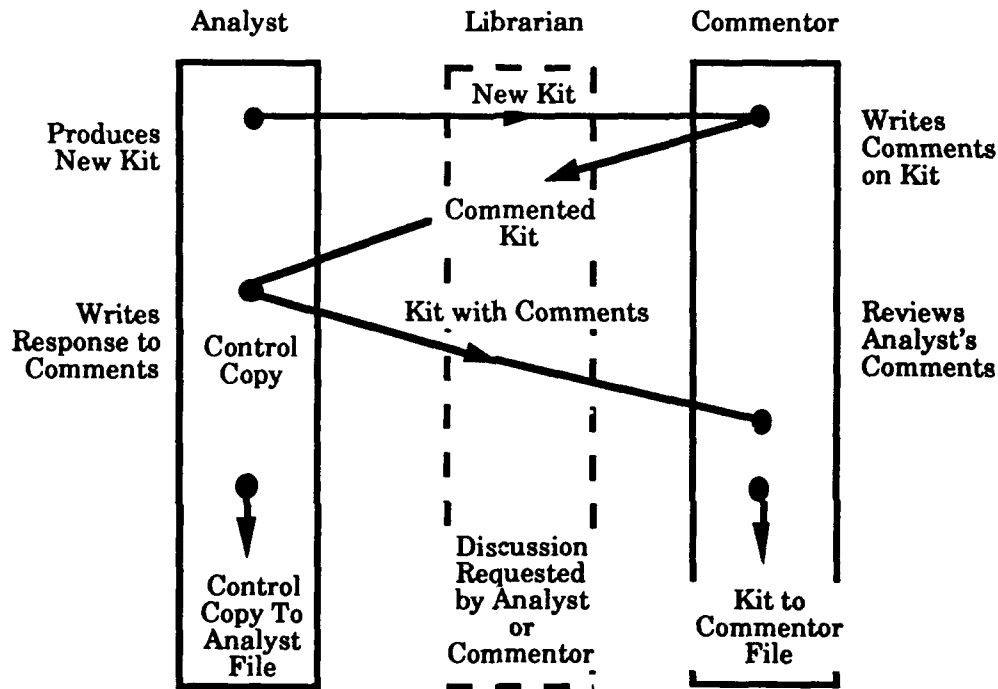


Figure 4-5
IDEF3 Kit Cycle

The following are the major steps in the IDEF3 kit review cycle.

1. The analyst assembles a kit (e.g., a pool kit, a scenario kit, an object kit, or a description kit with process flow diagrams and OSTN diagrams). The analyst retains one copy and gives one copy to the commentor for review.
2. The commentor reads and studies the contents of the kit within an agreed time period. The main purpose of this review is to determine whether the description is in compliance with the overall goals and context of the development effort. Comments will be made directly on the diagrams, other documents in the kits, and the cover sheet.

The kit with comments should be returned to the analyst by the date indicated on the cover sheet.

3. The analyst responds to the comments directly on the commentor's copy of the kit. The analyst may agree with comment, noting it on his working copy and incorporating it into the next version of the IDEF3 description. If there are disagreements, the analyst notes the points of disagreement on the kit and returns the kit to the commentor.
4. The commentor will read and file the returned kit if the analyst's responses are satisfactory. Otherwise, a meeting between the commentor and the analyst is arranged to resolve the disagreements.
5. This cycle continues until a mutually acceptable (to the analyst and commentor) IDEF3 description is produced.

The results of the IDEF3 kit cycle are an IDEF3 description to which the analyst and the commentor have contributed, and, if necessary, a list of issues that require management action. A valuable by-product of this review cycle is a recorded history of the review process.

Throughout the cycle, a project librarian handles copying, distribution, filing, and transfer of IDEF3 kits between the analyst and the commentor (see Figure 4-5).

4.5.2.3 Types of IDEF3 Kits

IDEF3 kits have a structure similar to those for other IDEF methods. They contain diagrams, text, descriptions, elaborations, and any associated material packaged for review and comments.

There are three types of IDEF3 kits:

1. Scenario kits address one scenario and all or part of its associated documentation. The following items may appear in a scenario kit.
 - A. Process flow diagrams and all associated UOB decompositions. Some of the review kits created

early in the development process may omit some of the decompositions.

- B. All available UOB elaborations and link specifications. Some of the scenario kits created early in the development process may omit some or all of these.
- 2. Object kits address one or more objects and all the associated OSTN diagrams, their descriptions, and their associated object state descriptions.
- 3. Description kits are created in the later stages of a development effort. A description kit is a compilation from the completed scenario and object kits for a given project. It contains all the scenarios in the IDEF3 description and their associated documentation. An approved description kit would represent one of the final deliverables in a development effort.

Scenario kits can provide any level of detail from a single-scenario process flow diagram to a complete process description that contains all elaborations and UOB decomposition diagrams. Description kits can also provide any level of completion; however, they will reflect the current status of the entire project as opposed to that of the single scenario of a Scenario kit. A more detailed description of Scenario and Description kits is given in Section 4.5.5.

4.5.2.4 Guidelines for Analysts and Commentors

4.5.2.4.1 Commentor Guidelines

No set pattern of questions and rules can be adequate for commenting, since subject matter, style, and technique vary so widely. However, guidelines do exist for improving quality. The major criteria for quality are: Will the document communicate well to its intended audience? Does it accomplish its purpose? Is it factually correct and accurate, given the bounded context? The following are overall guidelines for commenting.

- 1. Make notes brief, thorough, and specific. As long as the analyst understands that niceties are dropped for conciseness, this makes for easier communication and less clutter.

2. Use the @ notation to identify comments. To write @-note, check the next number off the NOTES list, number the note, circle the number, and connect the note to the appropriate part with a squiggle "~."
3. Make constructive criticisms. Try to suggest solutions rather than just making negative comments.
4. Take time to gather overall comments. These may be placed on the cover or a separate sheet. (Don't gather specific points on this sheet if they belong on the individual pages.) Agenda items for analyst/commentor meetings may be summarized. Make agenda references specific.

The time spent critiquing depends on several different factors: familiarity with what is being described, the number of times something has been reviewed, the experience of the commentor and analyst, etc. An IDEF3 kit returned to an analyst with no comments means that the commentor is in total agreement with the analyst. The commentor should realize that there is a shared responsibility with the analyst for the quality of the work.

4.5.2.4.2 Analyst/Commentor Interchanges

When a commentor returns an IDEF3 kit, the analyst responds by putting a "√" or "X" by each @-note. A "√" means the analyst agrees with the commentor and will incorporate the comment into the next version of the IDEF3 kit. An "X" means the analyst disagrees and requires a reason to be noted where the comment appears. After the analyst has responded to all comments, the IDEF3 kit is returned to the commentor.

After reading the analyst's responses, the commentor identifies remaining points of disagreement and requests a meeting with the analyst. This specific list of issues forms the agenda for the meeting.

4.5.2.4.3 Meeting Rules

Until comments and reactions are on paper, commentors and analysts are discouraged from conversing.

When a meeting is required, the procedure is as follows.

1. Each meeting should be limited in length.
2. Each session must start with a specific agenda of topics to be considered; discussions must not deviate from these topics.
3. Each session should terminate when the participants agree that the level of productivity has dropped and individual efforts would be more rewarding.
4. Each session must end with an agreed list of action items which may include the scheduling of follow-up sessions with specified agendas.
5. In each session, a "scribe" should be designated to take minutes and note actions, decisions, and topics.
6. Serious, unresolved differences should be handled professionally (i.e., documenting both viewpoints).

The result of the meeting should be a written resolution of the issues or a list of issues to be settled by appropriate managerial decision. Resolution can take the form of more study by any participant.

4.5.2.5 Contents of IDEF3 Kits

An IDEF3 kit is a technical document. It may contain diagrams, text, glossaries, decision summaries, background information, or anything packaged for review and comment.

4.5.2.5.1 General Guidelines for Kit Preparation

To avoid oversights, review the IDEF3 kit as if it were the only information available. Note any typographical errors and add points of clarification as brief notes on the IDEF3 kit itself. Glossary definitions for terms that appear in the IDEF3 kit should always be appended as support material.

Gather helpful materials and append these for the commentor's benefit. Never use this supplemental material to convey information which should

properly be conveyed by the diagram itself. Whenever possible, use the most natural means of communication to show details that are important for the reader in understanding the concepts. Combine all material with a completed cover sheet and submit to the librarian.

4.5.2.5.2 The Cover Sheet

The cover sheet distinguishes the material as an IDEF3 kit. The cover sheet has fields for analyst, date, project, document number, title, status, and notes. The following are the fields of an IDEF3 Kit Cover Sheet (see Figure 4-6).

1. IDEF3 Process Description/Document Description

Title: Should be descriptive of the IDEF3 kit.

Life-Cycle Step: "AS-IS" or "TO-BE" (does the kit contain a description of something that is or something that might be).

System: Acronym for System or Subsystem.

2. Project Information

Analyst: Name of person submitting the IDEF3 kit.

Date: Date sent to library.

Company: Name of the company submitting the IDEF3 kit.

3. IDEF3 Kit Information

Check Description kit, Scenario kit, or Object kit. Indicate document number assigned by the librarian.

4. Review Cycle

To be signed and dated after review by commentor and analyst.

5. Index/Contents:

List the Scenario, Decomposition, Object, and Object State (if relevant) names along with the C-number (discussed below) of each page of the document. An additional sheet called the IDEF3 Kit Contents Sheet (see Figure 4-7) is also filled out if necessary along with the Kit Cover Sheet.

DOCUMENT DESCRIPTION				PROJECT INFORMATION			KIT INFORMATION		REVIEW CYCLE		
TITLE:		LIFE-CYCLE STEP:		ANALYST:		DATE:		DESCRIPTION KIT		REVIEWER	
DEF METHOD:		SYSTEM:		COMPANY:		TASK NO.:		SCENARIO KIT			
				PROJECT NO.:				OBJECT KIT		ANALYST	
				PROJECT NO.:				SUPERSEDED OR REVISED DOCUMENT NUMBER		DATE	
COPY FOR		REVIEWERS		COPY FOR		REVIEWERS		LOG			
COM- MENT		NAME		COM- MENT		NAME		FILE			
PROJECT NUMBER		COMPANY		PROJECT NUMBER		COMPANY		AUTHOR			
										KIT CYCLE DATES	
										RECEIVED BY LIBRARY	
										KIT TO REVIEWER	
										COMMENTS DUE BACK TO LIBRARY	
										COMMENTS TO ANALYST	
										ANALYST RESPONSE DUE BACK TO LIBRARY	
										ANALYST RESPONSE TO COMMENTER	
										KIT CYCLE COMPLETE	
										COPYING INSTRUCTIONS	
										_____ copies of _____ pages = _____ total	
COMMENTS/SPECIAL INSTRUCTIONS											
Return Kit Library											
C-NUMBER _____ PAGE _____											
DOCUMENT NUMBER _____											
DESCRIPTION NAME:											

[illegible]

6. Comments/Special Instructions:

Any other information for the reviewers. This can also be used for special instructions to the librarian about handling the document. The library also uses this field for special instructions to the recipients of IDEF3 kits.

4.5.2.5.3 The Diagram Form

The diagram form, as shown in Figure 4-8, has minimum structure and constraints. The sheet supports only the functions important to the discipline of structured analysis. These are: 1) establishing a viewpoint, 2) cross-referencing between pieces of paper, and 3) documenting notes about the content of each sheet. The diagram form is a single standard size for ease of filing and copying. The form is divided into three major sections:

1. Working Information (top),
2. Message Field (center), and
3. Identification Fields (bottom).

The form is designed so that the working information at the top of the form may be cut off when a final "approved for publication" version is completed. The diagram form should be used for everything written.

Working Information

The following are the subfields that record working information.

1. Used At

This is a list of diagrams, other than the immediate context, which use this sheet in some way.

2. Analyst/Date/Project

This field documents who originally created the diagram, the date it was first drawn, and the project title under which it was created. The "date" field may contain additional dates, written below the original date. These dates represent *revisions* to the original sheet. If a sheet is re-released without any change, no revision date is added.

USED AT:	ANALYST:	DATE:	WORKING	REVIEWER:	VIEW:
	PROJECT:		DRAFT		
	NOTES: 1 2 3 4 5 6 7 8 9 10		RECOMMENDED		
			RELEASED		
	REV:	DATE:			

SCENARIO:	OBJECT:	DESCRIPTION NAME:	NUMBER:
DECOMPOSITION:			

Figure 4-8
IDEF3 Kit Diagram Form

3. Notes

This field provides a check-off for © notes written on the diagram sheet. As comments are made on a page, the notes are successively crossed out. This provides a quick check for the number of comments, while the circled number provides a unique reference to the specific comment.

4. Status

Four status classifications provide a ranking of approval: working, draft, recommended, and released.

Working: The diagram is a major change, regardless of the previous status. New diagrams are, of course, working copy.

Draft: The diagram is a minor change from the previous diagram and has reached some agreed-upon level of acceptance by a set of readers. Draft diagrams are those proposed by a project leader, but not yet accepted by the project team.

Recommended: Both this diagram and its supporting text have been reviewed and approved by the project team. This diagram is not expected to change.

Released: This page may be forwarded *as is* for final release or publication.

5. Reader/Date

This area is for the commentor to initial and date each form.

The Message Field

The message field contains the primary message to be conveyed. The field is normally used for diagramming, but the field can be used for any purpose (e.g., glossary, checklists, notes, sketches. The analyst should use no paper other than the diagram forms.

Identification Fields

The identification fields are as follows.

1. Title Field

The title field contains the name of the material presented on the diagram form. If the message field contains a diagram, the contents of the title field must precisely match the name written in the parent box.

2. Number

This field contains all numbers by which this sheet may be referenced.

C-number: The C-number is composed of two or three letters of the analyst's initials followed by a number sequentially assigned by the analyst. The C-number, placed in the lower left corner of the number field, is the primary means of reference to a sheet or form. Every diagram form used by an analyst receives a unique C-number. When an IDEF3 description kit is released, the C-number may be replaced by a standard sequential page number.

Page Number: An IDEF3 kit page number is written by the librarian at the right-hand side of the number field. This is composed of the document number followed by a number identifying the sheet within the document.

5.0 IDEF3 Development: Barbershop Example

The example presented in this section demonstrates the use of the IDEF3 method in a common place setting. Included in the example are common errors that a user may make. Moreover, justifications of the application of the process development steps are documented as a guide for the novice user. The example is a description of how a barbershop operates.

5.1 Define Purpose and Scope

Assume that the owner of a barbershop is interested in documenting the details of "what goes on in the barbershop" using the IDEF3 process flow description capture method. Thus, he hires an analyst to develop an IDEF3 process flow description of the shop. Assume that the need for this project stems from the owner's desire to record the description of the barbershop for potential employees. One benefit of applying the IDEF3 method in this situation will be that a new employee can quickly understand the operation of the shop from the IDEF3 process flow descriptions without the owner having to spend valuable time communicating this knowledge. In this example, the boundaries of the problem will be kept simply to the barbershop itself. The level of detail needed is specified to include only that information needed to clearly specify the workings of the barbershop to a new employee (e.g., barber or cashier). This purpose and context would be entered on the IDEF3 description summary form. At this stage of the process, the analyst would normally identify candidate scenarios and begin an IDEF3 scenario pool. The contents of this pool will be refined and maintained throughout the life of the project.

Note that in this example, only three modeling team roles are illustrated: 1) the analyst (the IDEF3 expert), 2) the domain expert (the barbershop owner), and 3) the client (also the barbershop owner). (Note that the domain expert

and the client are usually not the same individual.) The remainder of this section will refer to these individuals by their modeling team role names.

5.2 Collect Data

5.2.1 Interview Domain Expert and Acquire Initial Description

At this point in the development, the analyst will perform typical knowledge acquisition activities. The analyst will ask the domain expert, "How does your barbershop operate?" Suppose the domain expert answers the question with the following description:

The shop has two barbers, each of whom is given a chair to cut customers' hair. There are four chairs available for customers to sit in while they wait. Moreover, the shop has a cashier who works at a desk, and a magazine rack with magazines for use by customers. If a barber is free when a customer walks in, the customer sits in the barber's chair and has his hair cut. Otherwise, the customer has to wait until the barber is free. While waiting, the customer takes a magazine from the rack and reads it. However, if no barbers are free and all the waiting chairs are occupied, the customer gives a "disappointed look" and leaves. A customer whose haircut is complete will leave the shop after paying the cashier.

In practice, the completeness of the description provided by an interview will depend upon several factors.

1. The amount of time the domain expert is willing or allowed to devote to the interview.
2. The experience and domain-specific knowledge of the interviewer.
3. The domain expert's knowledge of the process that is being described.

During the interview with the expert, the analyst will acquire the initial description that may include written documentation about the process. The purpose of the description acquisition is to represent how the system *actually*

works, rather than how the domain expert *thinks* the system works (or how the domain expert thinks the system *should* work). Therefore, the analyst needs to correlate facts captured in the interview process with first-hand observations of the process. The analyst also must avoid completing the description with his own (often preconceived) knowledge about how the system ought to work. Thus, it is important that both the analyst and the domain expert understand that descriptions are often partial in nature and curb the tendency to force them to idealized completions.

5.2.2 Analyze Description for Data Identification

Once the interview is over, the analyst needs to carefully study the notes and observations he has recorded. The purpose of this analysis is to identify the objects, activities, facts, and constraints that occur within the description. This step can be conceived as a list making process.

When describing processes, individuals often focus on the key objects in the process and their roles in the process before actually describing the events or activities that occur during the process. The following is a list of objects that were identified in the barbershop description.

Customer	Barber	Haircut Needed
Barbershop	Waiting Chairs	Magazine
Barber's Chairs	Cashier	Magazine Rack

It is important that the analyst explicitly record the list of objects in the IDEF3 object pool for the following reasons.

1. He may omit some of the objects at a later stage in the description capture process.
2. This list of objects from the first analysis often contains the primary objects in the process. Primary objects are those objects important enough to warrant the creation of OSTN diagrams.

After identifying objects, the interview notes are examined to determine the activities/processes that occur in the barbershop. The important activities are candidates to be represented as UOBs (activities, actions, or processes) in the description. However, at this stage in development the sequence of the activities is not important. The primary goal is to list the candidate UOBs (as shown in the list below). These candidate UOBs would be listed in the IDEF3 UOB pool. It is likely that the list of UOBs is incomplete; however, this is not a matter of much concern at this stage.

1. Customer Arrives
2. Customer Sits in Barber's Chair
3. Customer Waits in Chair
4. Barber Cuts Customer's Hair
5. Customer Leaves Dissatisfied
6. Customer Pays Cashier
7. Customer Leaves
8. Customer Reads Magazine

The final step in the analysis of the interview involves identifying and listing facts, and identifying the constraints relevant to the processes described by the domain expert. The facts are assertions made about the objects. Constraints are distinguished conditions that are known to hold between the objects within a process or between the processes themselves. To identify the occurrence of constraints, look for *negative* terms such as *not*, *never*, or *no* within the recorded verbal description (as shown in the following list). The list of facts and constraints is likely to be incomplete early in the development. Further interviews or conversations with the domain expert will aid in making the lists more complete.

- | | |
|-------------------------------------|-------------------------------------|
| 2 Barbers | 2 Barber chairs |
| 4 Waiting Chairs | Barber can be cutting hair or free. |
| No barbers or chairs are available. | |

5.3 Formulate Process Flow Descriptions

Once the initial task of identifying objects, activities, facts, and constraints nears completion, the stage is set to formulate the IDEF3 process flow descriptions. The observations recorded in the interview process are used as the basis for developing the process flow descriptions. The candidate UOBs listed in the data analysis phase will be used in this step to construct the UOBs. The facts and constraints identified during the analysis of the interview(s) will be used in the construction of the UOB elaborations. The process flow development occurs in two major stages, the construction of 1) UOBs in correct sequence and 2) UOB elaborations.

5.3.1 Identify the Sequence of Units of Behavior in the Diagram

The process of identifying the UOBs and specifying the precedence between them occurs in a stepwise manner.

Step 1. The first step is to identify the leftmost UOB in the process flow, the UOB *Customer Arrives*.

Step 2. The second step is to identify the next UOB. In this example, three UOBs are possible: *Customer Sits In Barber's Chair*, *Customer Waits in Chair*, or *Customer Leaves Dissatisfied*.

The second step implies a split in the process flow, indicating the need to use a fan-out junction to represent the diverging flow. The analyst must determine the junction type that initiates the split. In this example the customer can perform only one of the three alternative activities; therefore, an XOR junction is used. The analyst may find it useful at this stage to create the partial diagram shown in Figure 5-1.

If a split in the process had not occurred, the development would have continued with the sequential drawing of UOB boxes until a split did occur. After a split, each process path is developed separately. These process paths may or may not converge within the context of the given description. The order in which the process paths are developed is a matter of preference.

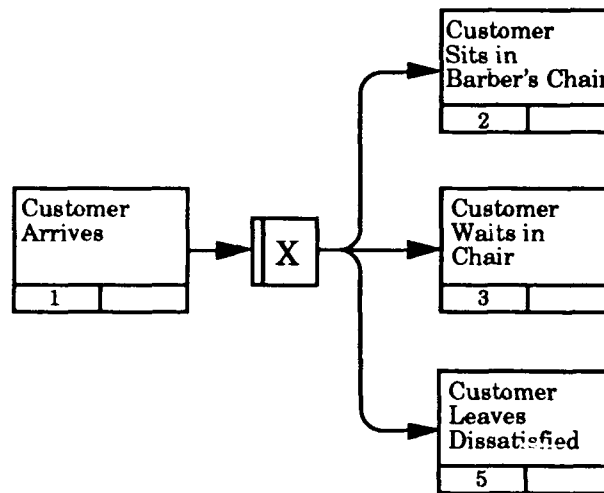


Figure 5-1
First Steps in Scenario Description

Step 3 The next step is to develop the path that begins with UOB 2. This path continues sequentially with the UOBs *Barber Cuts Customer's Hair*, *Customer Pays Cashier*, and *Customer Leaves*. These UOBs result in the partial diagram shown in Figure 5-2.

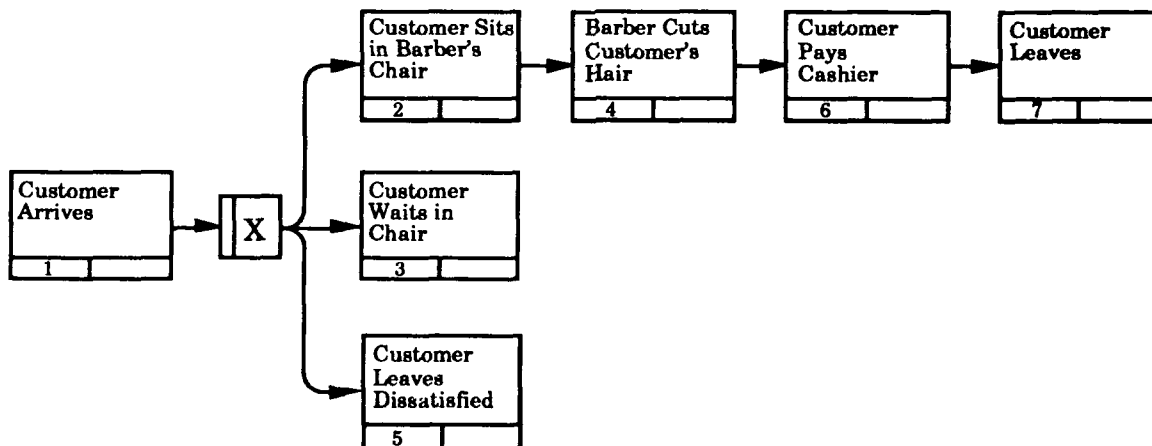


Figure 5-2
Diagram with First Path Complete

Step 4 The fourth step is to complete the remaining two paths in Figure 5-2, resulting in the process flow diagram shown in Figure 5-3. Note that the UOBs retain the numbers assigned as they were placed in the activities list.

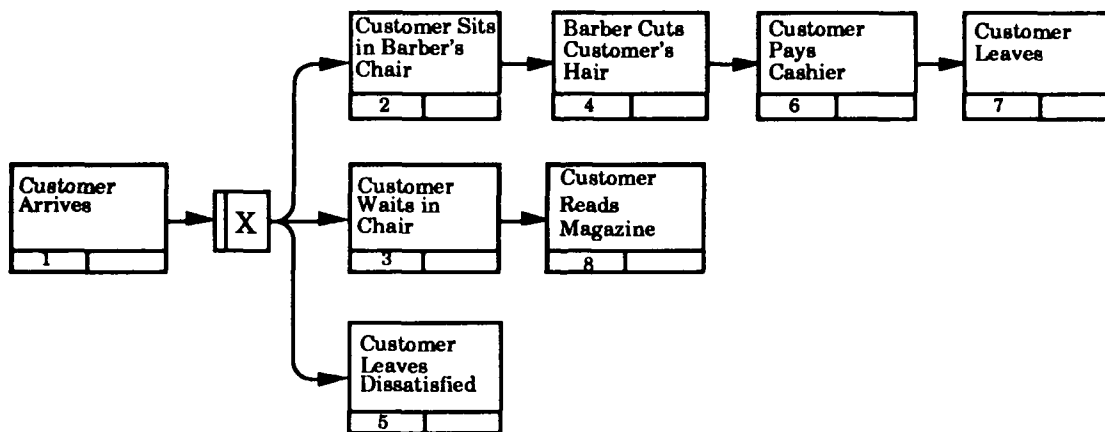


Figure 5-3
Diagram Near Completion

Step 5 When the diagram illustrated in Figure 5-3 is finished, all the activities in the list of potential UOBs have been “used up” to create UOBs. However, in the description provided by the domain expert, it was implied that the waiting customer will eventually get a haircut. After the customer has waited, the next activities in the description will include *Customer Sits in Barber's Chair*, *Barber Cuts Customer's Hair*, *Customer Pays Cashier*, and *Customer Leaves*. To ensure these actions are represented in the process flow diagram, a referent is included in the diagram following the UOB *Customer Reads Magazine*. This produces the flow diagram shown in Figure 5-4.

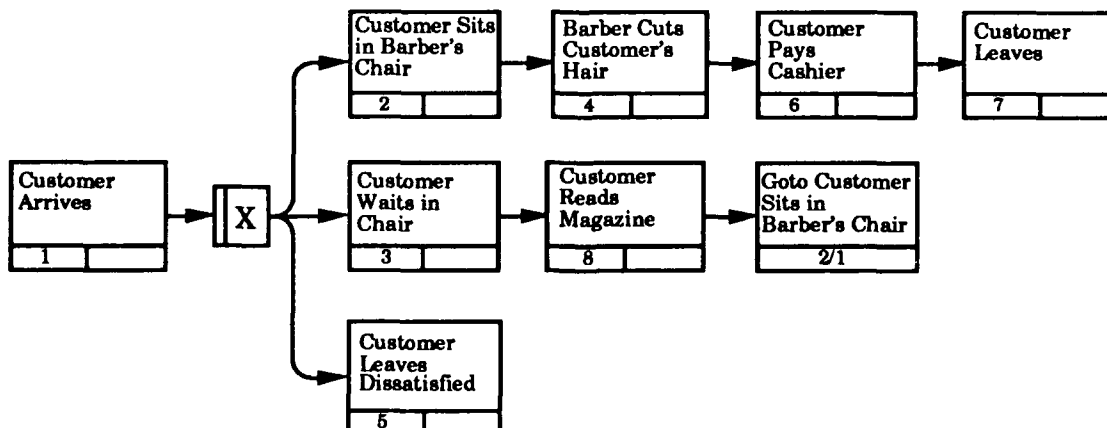


Figure 5-4
Complete Flow Diagram Before First Review

5.3.2 Analyze Data for Unit of Behavior Elaborations

After the process flow diagram has been completed, elaborations must be added to each UOB as shown in Figures 5-5 and 5-6. In the initial attempt, these may be somewhat incomplete. One reason for this may be that the primary focus of the analyst in the first interview is on the objects and activities. This is particularly true in the development of either a description for a process with which the analyst was unfamiliar or a description of a large, complex process.

When the analyst is familiar with the process type (as in this barbershop example), he will be able to obtain more information about the particular process in the first interview. The analyst's questions would reflect this familiarity and he could determine in the first interview how the process differs from other systems of this type. In developing the elaborations, the analyst again needs to avoid allowing his knowledge of the system type to influence the information placed in the elaborations.

The order in which the elaborations are developed is not important. It may often be useful to develop elaborations in parallel with the rest of the process flow diagram, because, in some situations, this may aid the analyst in structuring the diagrams. However, for this example, the initial elaborations were developed after the rest of the process flow diagram was complete. The elaborations that resulted are shown in Figures 5-5 and 5-6. Note that for brevity in this example, we have not included the constraint lists in these elaborations. Recall that each link in the process flow diagram would generate a constraint entry in the elaborations of each linked UOB.

5.3.3 Review Process Flow Description(s) with Domain Experts

The state of completion of the elaborations will depend on the depth of the first interview and the amount of information obtained during it. The analyst should attempt to make both the process description and the elaborations closely reflect the domain expert's view (as obtained in the interview).

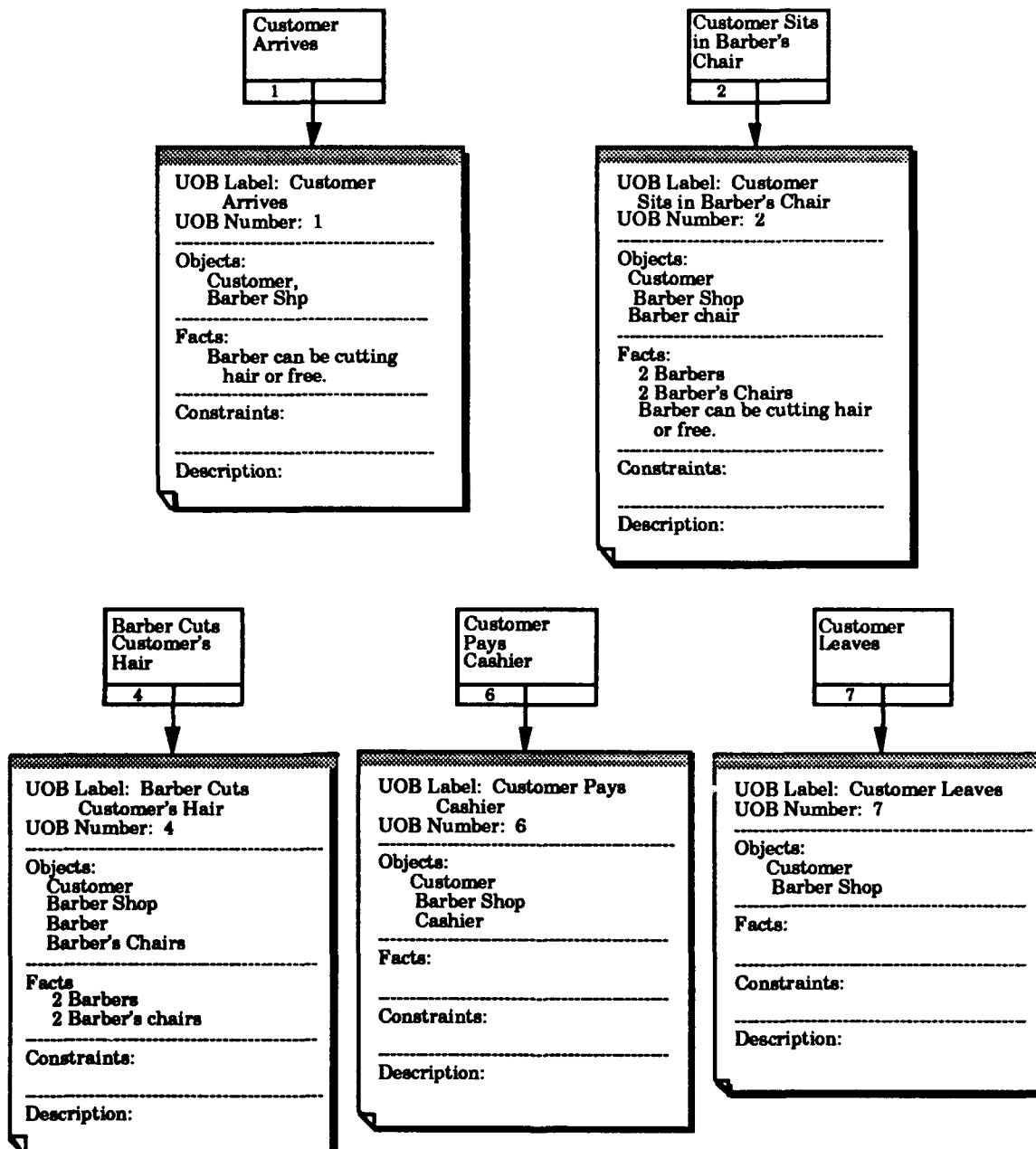


Figure 5-5
Elaborations

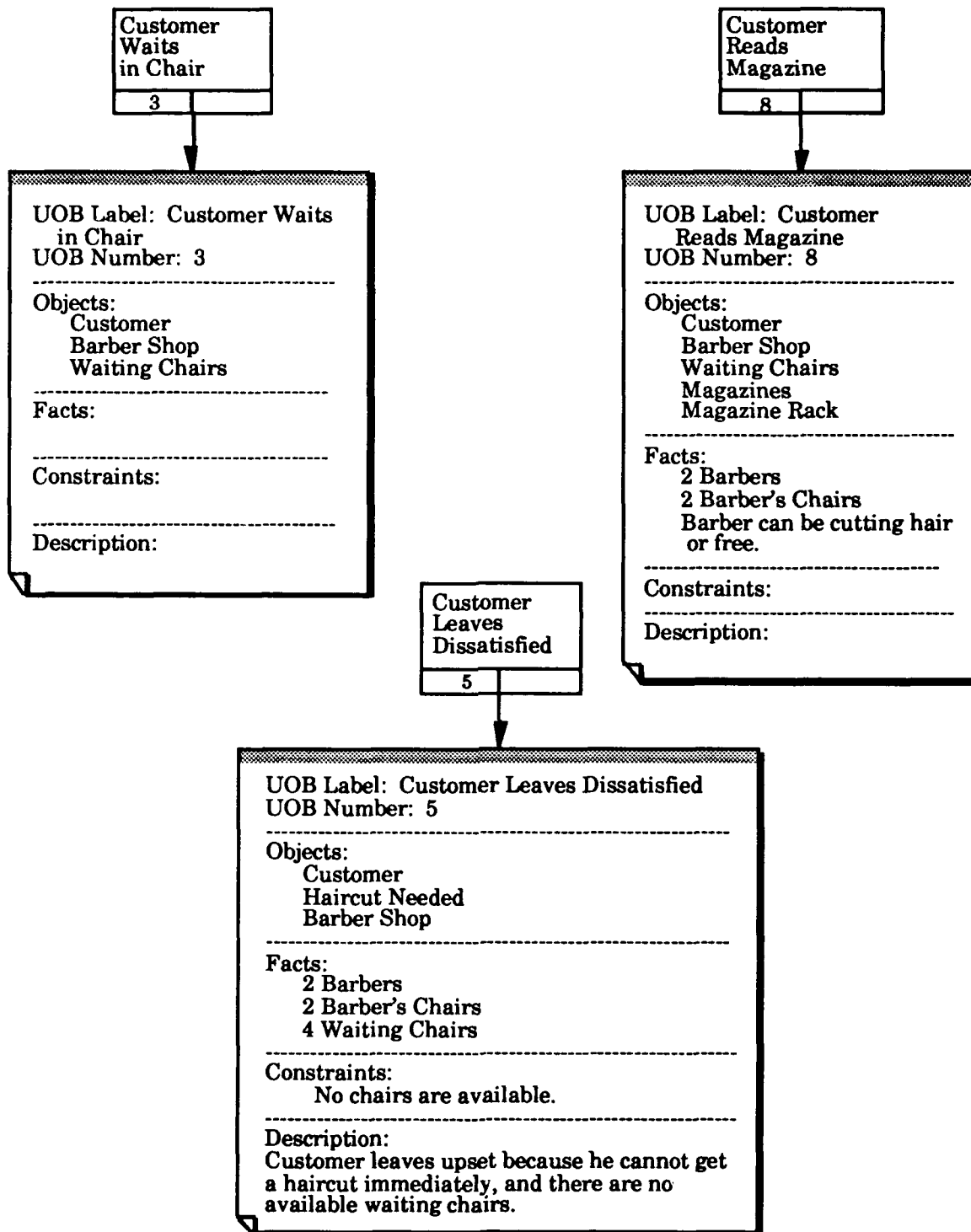


Figure 5-6
Elaborations (Continued)

In this example, the analyst has made the description as complete as possible and will return to the domain expert for an evaluation of the description. In this interview, the structure of the diagram would be evaluated to confirm that it communicates the expert's knowledge about the scenario. The correctness of the diagrams along with the elaborations will be confirmed in this process. The review may indicate that some changes need to be made to the captured description. This can take the form of either additional objects, activities, facts, and constraints or modifications and deletions to the original lists.

After reviewing the IDEF3 description with the barbershop owner (the domain expert in this example), the analyst made the following observations which required changes in the IDEF3 process flow description.

1. Some customers have a favorite barber.
2. If a customer has a favorite barber, he will wait until that barber is available before getting his hair cut.
3. Customers waiting for haircuts are served on a first-come, first-served basis.
4. After getting a haircut, the customer examines the cut, pays the cashier, and, if dissatisfied with the haircut, gives a disappointed look before leaving.
5. In this interview, the analyst needs to know: "How does the customer decide between the processes indicated by UOB 2, UOB 3, and UOB 5?" as shown in Figure 5-4. The following is the response from the domain expert as follows.

The first thing that the customer does when he comes into the barbershop is look for a free barber. If 1) free barber is his favorite or 2) he has no preference for a barber, the customer sits in the barber's chair. Otherwise, he sits in a waiting chair. If there are no free chairs at that time, he will give a disappointed look and leave.

After the review and interview, the new data is evaluated and the lists are updated. The additional data is incorporated into the description in the following manner.

The following are the Object List Additions:

Favorite Barber.

Dissatisfied Customer.

Satisfied Customer (Customers will be either be dissatisfied or satisfied after their hair is cut.)

The following are the UOB Pool additions (action or process additions):

Inspect haircut (UOB 9).

Look for free barber (UOB 10).

Leave Dissatisfied (UOB 11).

Note that the numbering of the UOBs continues sequentially.

The following are the Facts and Constraints (Additions):

Queuing discipline is first-in, first-out (FIFO).

Some customers have a favorite barber.

Customers will wait for their favorite barber to be free before getting a haircut.

Customers may be either satisfied or dissatisfied with their haircut.

The additional data and changes suggested by the domain expert are incorporated into the process flow description. This resulting process flow diagram is illustrated in Figure 5-7. The activity represented by UOB 11, *Customer Leaves Dissatisfied* may at first appear identical to the activity represented by UOB 5, *Customer Leaves Dissatisfied*.

If the behavior represented by UOB 11 is the same as that represented by UOB 5, UOB 11 must be replaced with a referent to UOB 5. This determination cannot be made, however, without examining the objects, facts, and constraints associated with UOB 11 and UOB 5.

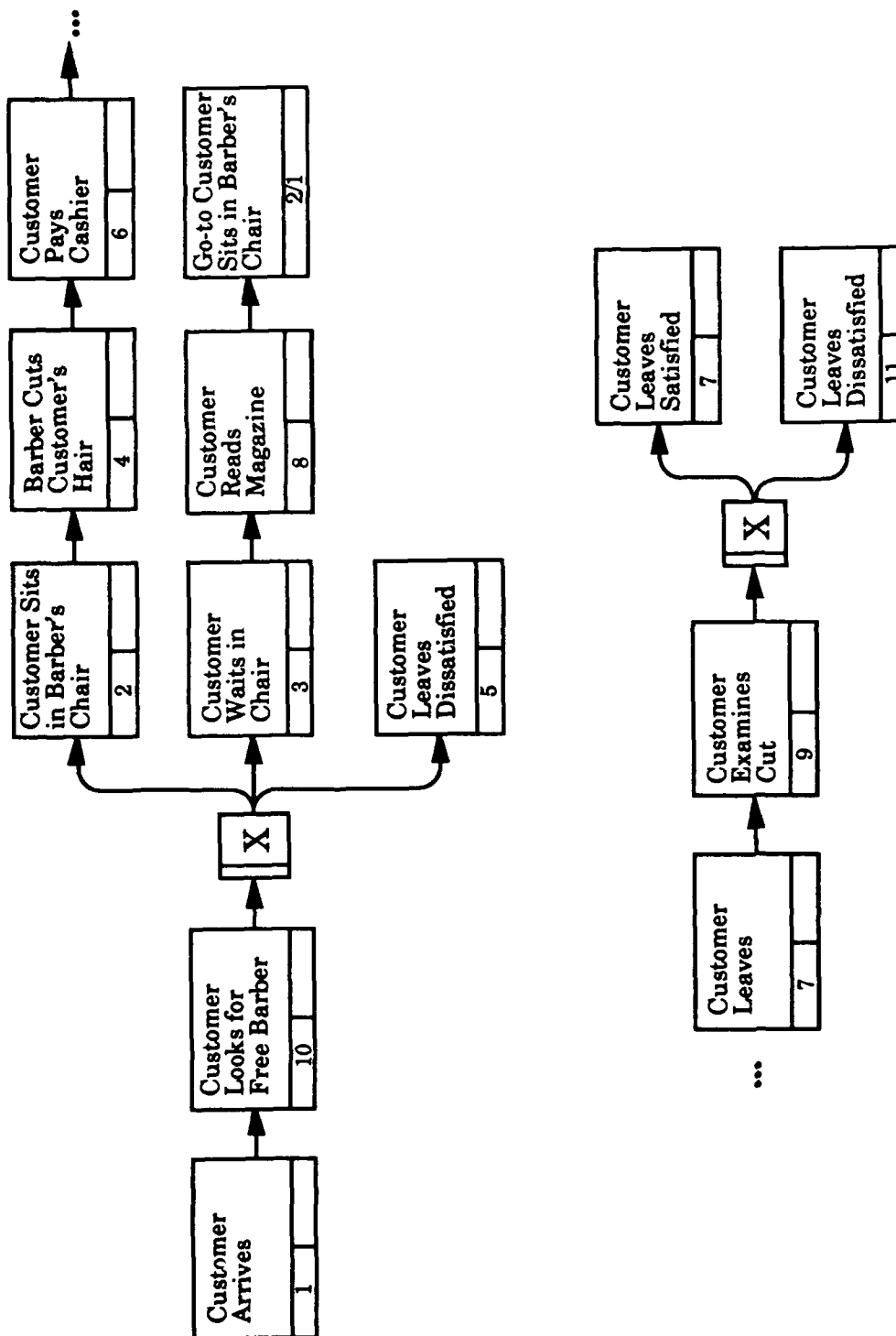


Figure 5-7
Final Process Flow Diagram

Examination of UOB 11 (see Figure 5-8) indicates objects, facts, and/or constraints that do not apply in UOB 5. (See Figure 5-6 for the comparison.) Because the objects, facts, or constraints are different, a new UOB must be created. (The primary difference is that in UOB 11 the customer has a bad haircut whereas in UOB 5 the customer was unable to get a haircut.) The new UOB 11 is labeled *Customer Leaves Dissatisfied*, to differentiate it from UOB 5.

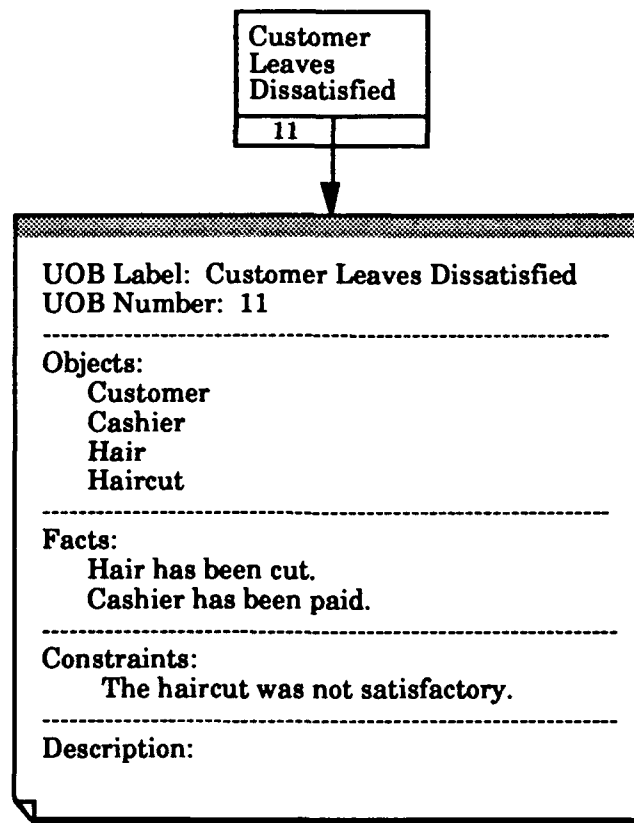


Figure 5-8
Elaboration for Customer Leaving Dissatisfied

In the final diagram (see Figure 5-7), the logic associated the junction J1 needs a more detailed explanation (see Figure 5-9). This is accomplished by attaching an elaboration, via a referent, to the junction J1. Note that within the elaboration form, the label field simply identifies the type of junction. The number field is the number attached to the junction (J1). An elaboration

is attached to a junction to clarify the decision logic associated with the junction. In the case of an XOR junction, the junction elaboration allows the analyst to fully describe the rules that determine the choice of a particular path out of the junction.

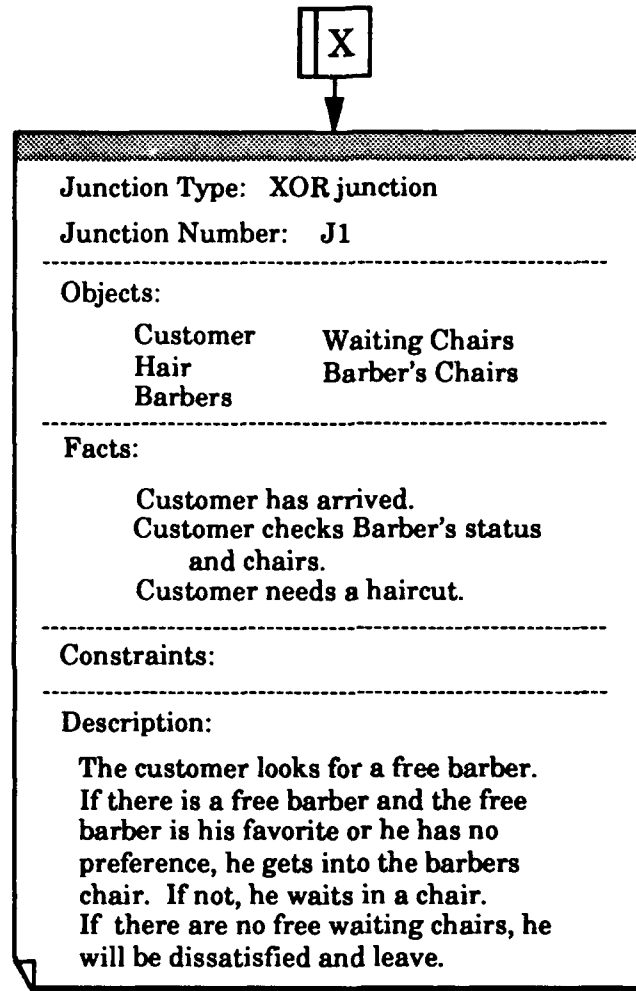


Figure 5-9
Elaboration Attached to a Junction

Another addition to this process flow is the use of a link specification for one of the links (see Figure 5-10). This link specification may not have been entirely necessary in a situation this simple; however, it is provided to illustrate how a link specification can be associated with a particular link. The link is assigned a number that allows a reader to associate a particular

link specification with the correct link. This link specification will contain relevant information to the link between the two participating UOBs.

Link Specification Document	
Link Number: L1 Link Type: Precedence	
Sources: Customer waits in chair. (UOB 3)	Destinations: Customer reads magazine. (UOB 8)
Objects: Customer Needing Haircut Magazine Rack Magazines Waiting Chair	
Facts: Magazine rack has magazines. Customer has waiting chair.	
Constraints: Either no barber is free or customer's favorite barber is not free.	
Description (text): The customer is either waiting for a free barber or for his favorite barber to get free.	

Figure 5-10
Link Specification Document

5.4 Formulate Object State Transition Network Diagrams

To provide a detailed characterization of the objects that participate in a process, it is useful to construct the OSTN diagrams. These are typically developed only for the important objects of the process flow description. OSTNs provide a different view of the process being described, i.e., an object-centered view.

Suppose that, in the barbershop example, the customer's *hair* is the important object. It may be useful to conceptualize the hair object as transitioning through several states in the process being described. Figure 5-11 shows the OSTN diagram for the hair. Initially, the hair is in the state labeled *Hair That is Too Long*. From this state, it progresses to the state *Hair Cut* after which it transitions to one of two states, *Hair That Looks Good* or *Hair That Looks Bad*. In addition to the diagram in Figure 5-11, an OSTN will have associated with it a description and each object state within the OSTN will each have a separate descriptions.

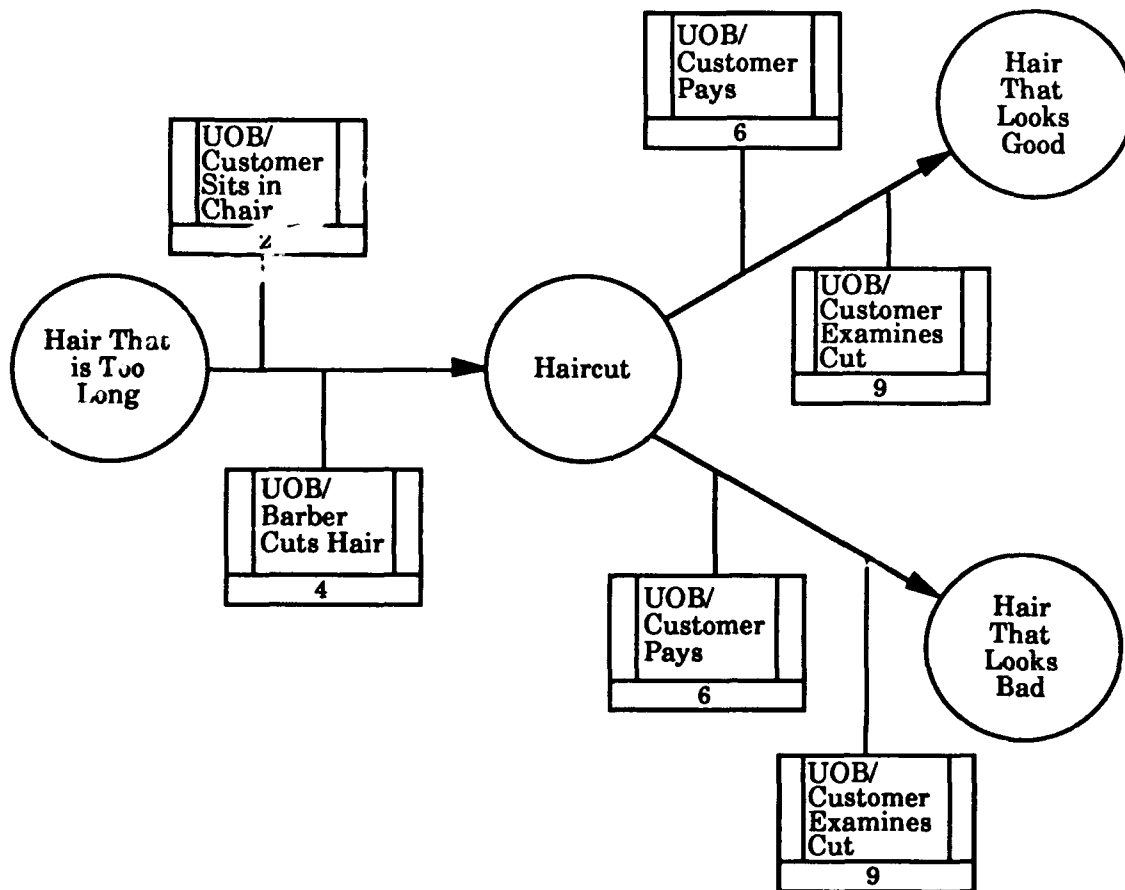


Figure 5-11
Object State Transition Network Example

6.0 Understanding IDEF3 Process Descriptions

The main purpose of an IDEF3 process flow description is to provide an accurate representation of how a particular system or organization works. An IDEF3 process flow description captures the factual descriptions of the process flow and the object state transitions associated with a particular scenario. Reviewers of IDEF3 descriptions may not create them, but must validate the facts in the descriptions. Readers of IDEF3 descriptions may need to acquire knowledge from descriptions that others have created. For the reviewer and reader, the procedure for reading and understanding IDEF3 process flow descriptions is addressed in this section.

An IDEF3 process flow description is usually read starting with the leftmost UOB of a scenario. Conventionally, a description is read from left to right. To obtain an overview of the described system, a *walkthrough* of the UOBs is performed. During this walkthrough, the reader notes precedence relationships and the logical layout of the UOBs. Such a reading will provide a general understanding of the system. Further details of a description may be obtained by reading each UOB and link with their elaborations or descriptions. A comprehensive understanding of the IDEF3 process flow description can be obtained by systematically studying the logic encoded within the descriptions.

6.1. Description Reading Steps

The facts collected about a system are structured in the IDEF3 process flow description. The approach to reading a description is usually dependent upon the reader and the amount of information the reader expects to derive.

Owing to the individualized nature of the description reading process, it is difficult to express the process in a strict algorithmic format. For example, some people prefer to first scan the diagram, then break it up into logical

pieces that are easier to understand. Each piece is subsequently analyzed with the goal of understanding the relationships between the UOBs and links within selected portions of the description. Once the meaning of the smaller pieces of the description are understood, the larger picture becomes evident by taking into account the junctions and their associated logic. The approach to reading a diagram can be summarized as follows.

1. Carefully read the statement of purpose, the statement of scope, the objective of the scenario being described, and the viewpoint of the IDEF3 process flow description.
2. Scan the UOBs, links, and junctions from left to right to gain a general impression of what is being described and generally understand the flow logic of the scenario.
3. Partition the diagram from left to right into logical structures of UOBs, links, and junctions. Logical structures are combinations or structures of UOBs, links, elaborations, and junctions that are conceptually or logically complete. These logical combinations will be process paths and may themselves contain logical structures or substructures. To achieve a better understanding of the description, these structures and substructures may have to be partitioned in the same manner that the overall diagram was partitioned.
4. Starting with the first structure on the far left of the description, read the diagram from left to right using the following guidelines.
 - A. Read the UOBs and their elaborations.
 - B. Examine links and note the information found in the link specifications.
 - C. Study all referents within the bounds of the selected structure.
 - D. Conduct a mental walkthrough of the description, one basic structure at a time.
 - E. When junctions are encountered, follow the paths noting the conditions under which a path will be selected and those under which other paths will be followed.
 - F. Check to see whether the placement of the paths is consistent with the logic of the description.

For more casual readers, a simpler approach is often used. This simpler approach is described in the next section.

6.2 Quick Reading of IDEF3 Process Descriptions: An Example

More casual readers of an IDEF3 process flow description will follow a similar process to that described in Section 6.1. However, they can expect that as they gain experience in the process, their approach will become personalized. An example approach for reading a scenario diagram is described in the following steps. This outline for reading a diagram would be repeated, with few modifications, for all decompositions of any UOBs. Generally, the UOB decompositions are read after the scenario diagram has been read and understood.

6.2.1 The Big Picture

A crucial step in the description-reading process is to understand the big picture relevant to the described real-life situation. This big picture can be gained by reading and understanding the statement of purpose, statement of scope, objective of the scenario being described, and viewpoint of the IDEF3 process flow description. These parts of the description bind the scope of the diagram and tell readers (particularly those familiar with the process being described) what to expect in the top-level diagram. They also indicate the level of detail anticipated.

6.2.2 Scan the Diagram

Readers should become familiar with the scenario by scanning the diagram from left to right. This involves becoming familiar with the UOBs, links, and junctions displayed in the diagram. This activity is not an in-depth study of the diagram; rather, it provides readers a general impression of what is being described and an overall understanding of the logic flow within the scenario.

6.2.3 Understand the Scenario

In this step, readers gain a detailed understanding of the process flow diagram associated with a scenario (or a UOB decomposition). This is the part of the communication process that is most individualized and requires the most time. It is helpful to partition the diagram into understandable pieces. The partitioning procedure described here is based almost entirely on the structure of the diagram. The example IDEF3 diagram in Figure 6-1 is partitioned as shown in Figure 6-2.

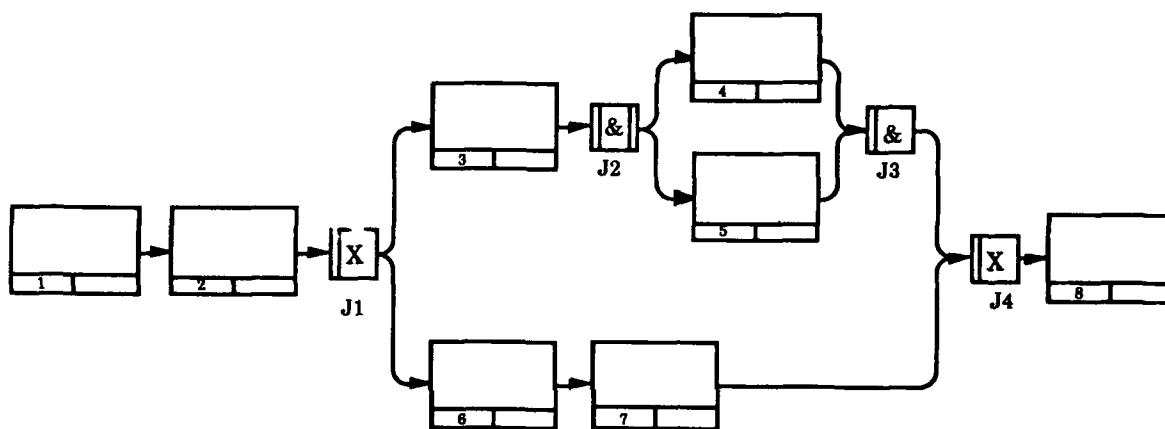


Figure 6-1
Example IDEF3 Diagram

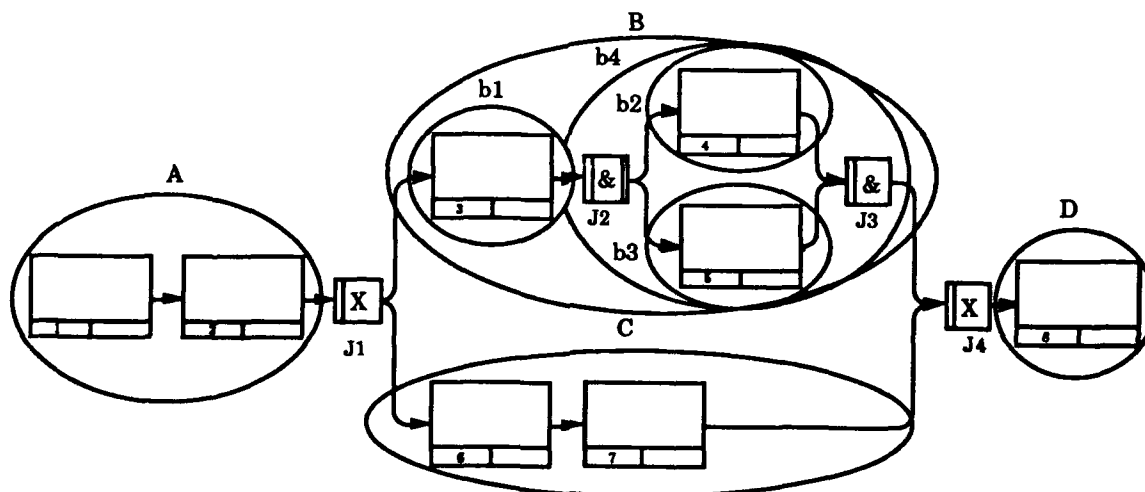


Figure 6-2
Partition the Diagram

In Figure 6-2, the diagram is first partitioned into four major structures: A, B, C, and D. These structures represent four process paths. An examination of these structures reveals that B can be further partitioned into substructures b1, b2, b3, and b4. No further breakdowns are possible; therefore, analysis of the individual structures can begin.

Numbering the structures 1 through 8 (see Figure 6-3) and starting with structure 1 on the far left of the description, readers will typically proceed from left to right and perform the following activities:

1. Read the UOBs and their elaborations.
2. Examine links and note the information found in the link specifications.
3. Consider all referents within the bounds of the selected structure.

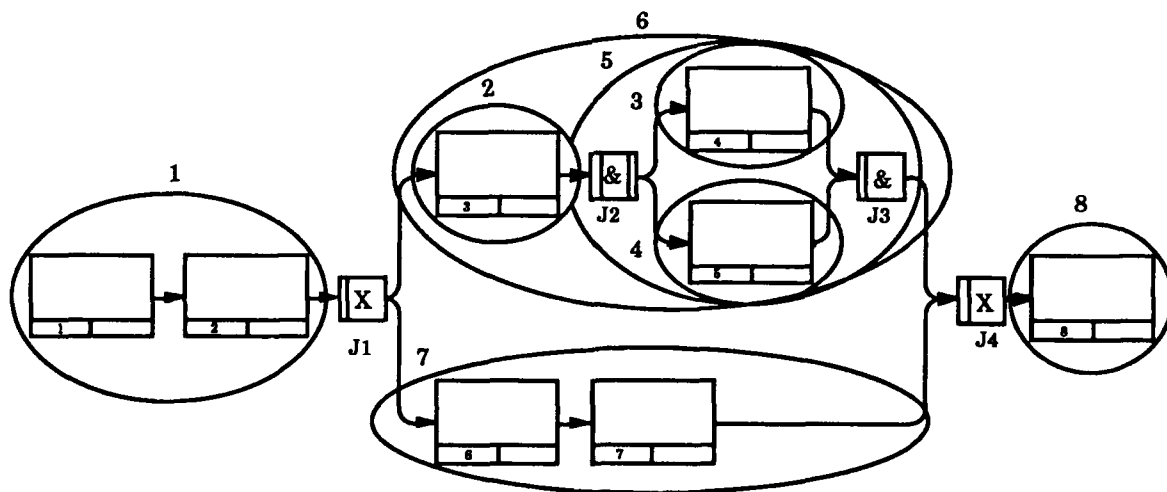


Figure 6-3
Analyzing the Structures

After understanding structure 1, the reader will study either structure 6 or structure 7. Note that the junction J1 is not immediately considered at this stage. Starting with structure 6, each of the substructures 2 and 5 (themselves structures) will be analyzed. The analysis of structure 2 means that one UOB and its elaboration must be studied. Structure 5 is a complex

structure which will be first subdivided into the structures 3 and 4. After completing the study of structures 3 and 4, structure 6 is analyzed in its entirety. This process involves understanding the logic of the structure that includes the J2 fan-out junction from structure 2 to the structures 3 and 4. To understand junction J2, readers examine the two paths leading from it and notes the conditions of flow to these paths. In general, the logic of a junction is analyzed by following all the paths leading in or out of it, and noting the conditions under which each path will be selected. The study of structure 5 is completed by analyzing the logic of the fan-in junction J3.

Structure 7 is analyzed by proceeding from left to right as follows:

1. Read UOBs 6 and 7 and their elaborations.
2. Reading from left to right, examine links and note the information found in the link specifications.
3. Consider any referents within the bounds of the selected structure.

After completing the analysis of structure 7, reading of the description will continue with the analysis of fan-out junction J1. The reader would perform a walkthrough of the process starting from structure 1, noting the conditions under which the flow would branch at the junction and the conditions governing each fan-out path.

The next descriptive element of the diagram to be analyzed is the fan-in junction J4 that enables merging of the process paths which are emerging from structures 6 and 7. Readers would do a walkthrough that involved analyzing the logic of junction J4, noting the conditions under which the two process flow paths converge.

Finally, structure 8 is analyzed by reading UOB 8 and its elaboration, and considering any referent that may be attached to it. After this, readers may want to do a complete walkthrough of the entire diagram. This will involve starting again at the left end of the diagram and continuing through to structure 8 considering all the junctions.

7.0 Practical Guidelines for Using the IDEF3 Method

In this section, practical guidelines for using the IDEF3 method are outlined.

7.1 How to Construct Valid IDEF3 Diagrams

IDEF3 is a method used to capture descriptions about the real-world in a structured, intuitive manner. The IDEF3 language was designed to support the capture of partial knowledge about the operation of a system. The method provides the user considerable freedom in terms of how these descriptions can be structured; the syntax of the language imposes only a few restrictions on the possible diagram configurations that are considered valid. These restrictions, or rules, will ensure that the syntax and semantics of the constructed descriptions capture the intent of the user to the fullest extent. Moreover, these validation checks try to enforce standardization between the potential users of the language in a manner that enhances the utility of the method as an unambiguous means of communication.

Validation is the process of checking and ensuring that a valid IDEF3 process description is constructed. There are three types of validation, *syntactic*, *semantic*, and *model theoretic*. Syntactic validation activities relate to ensuring that the IDEF3 diagram constructed conforms to the syntactic rules of the IDEF3 language. Syntactic validation is sometimes referred to as *verification*. Semantic validation activities relate to ensuring that the IDEF3 diagram statements accurately capture the assertions of the domain expert. Model theoretic³ validation activities check the consistency and completeness of a description against a *formal theoretical framework*.

³ In this context, the term *model theoretic* refers to a logic and mathematical idealization of process behavior. Hence, model theoretic validation compares the results of applying the

Potential IDEF3 users need to be aware of the difference between an AS-IS and a TO-BE description. An AS-IS description is a collection of assertions about a process or organization as it currently operates. A TO-BE description is a collection of assertions about a system that is to be developed. Model theoretic errors in an AS-IS system description may simply indicate limits in the knowledge of how the AS-IS system works. However, model theoretic errors in a TO-BE system description indicate potential errors in the system design. That is, while it is possible that AS-IS system descriptions contain inconsistencies, the developers of systems need to ensure that TO-BE system descriptions are free of inconsistencies.

7.1.1 Model Theoretic Validation Rules for IDEF3 Process Descriptions

The IDEF3 syntactic validation rules were presented in Section 3 of this document. The semantic validation is accomplished by the kit review process described in Section 4 of this document. This section will present the model theoretic rules for IDEF3 process descriptions. These rules are formulated in terms of additional syntactic rules. Thus, the model theoretic validation rules include those rules presented in Section 3 as well as additional constraints that allow for a logical analysis of the resulting diagrams. The following is a list of the primary model theoretic validation rules which enable the model theoretic validation of an IDEF3 description.

1. Every description must have a scenario name.
2. Every description and scenario requires a statement of need, purpose, and scope.
3. There can be only one *leftmost point* for every scenario (other than a decomposition) and every decomposition. A leftmost point is either a UOB or a junction.

method against this idealization. This model theoretic idealization is described in detail in (Menzel, 1991).

4. There can be only one *rightmost point* for every decomposition. A scenario which is not a decomposition can have more than one rightmost point.
5. Two or more *disconnected* graphs are not allowed within any decomposition.
6. No IDEF3 element (UOB or junction) can be directly linked back to itself.

The following is a list of UOB and UOB decomposition rules.

1. Every UOB must have a label (name).
2. Every decomposition must have a name.
3. The sibling decompositions of a UOB are numbered sequentially as they are created. Decomposition numbers are not unique within the description, scenario, or diagram; they are unique within their sibling set.
4. Multiple precedence links cannot lead out from a UOB. The need to create multiple links out from a UOB indicates that a fan-out junction is required.
5. Multiple precedence links going into a UOB *are* allowed; however, the semantics for interpreting the timing and logic of these links must be specified in the elaboration of the concerned UOB.

Many junction-and-link-related checks relate to the identification of *structures*. A structure is any logical syntactical combination of UOBs, links, and junctions. In Figure 7-1, A, B, and C are structures. Structures can be complex or simple. A complex structure is the portion of an IDEF3 diagram between a fan-out junction and its corresponding fan-in junction (e.g., A). A simple structure is any segment of an IDEF3 diagram without junctions (e.g., B and C). A simple structure can be part of a complex structure. Many junction and link rules relate to ensuring the syntactic validity of complex structures. The following is a list of syntactic rules which enable the model theoretic validation of an IDEF3 description for junction and link combination:

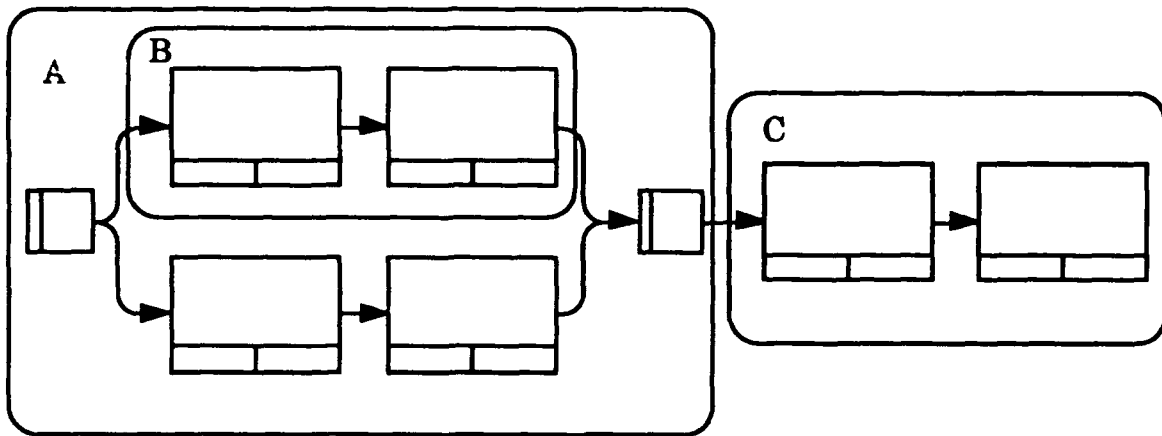


Figure 7-1
Structures

1. Every fan-in junction requires a matching fan-out junction.
2. Loops from within a complex structure to any point *outside* the structure are not allowed. For example, it is incorrect to create a link from structure A in Figure 7-1 to structure C.
3. A fan-in AND junction cannot be matched with a fan-out OR junction.
4. A fan-in AND junction cannot be matched with a Fan-out.
5. A fan-in XOR junction cannot be matched with a fan-out AND junction.
6. Every fan-in junction must have two or more incoming precedence links.
7. Every fan-out junction must have two or more outgoing precedence links.

7.2 Some Common Errors and Guidelines to Constructing IDEF3 Diagrams

7.2.1 Fan-out XOR Junction Followed by a Fan-in AND

An XOR fan-out junction may not be followed by a structure-closing fan-in AND junction. The violation of this condition would represent an attempt to

describe a *model theoretically inconsistent* situation. In other words, all attempts to instantiate such situations in the real-world would certainly fail. To illustrate, consider the IDEF3 diagram shown in Figure 7-2.

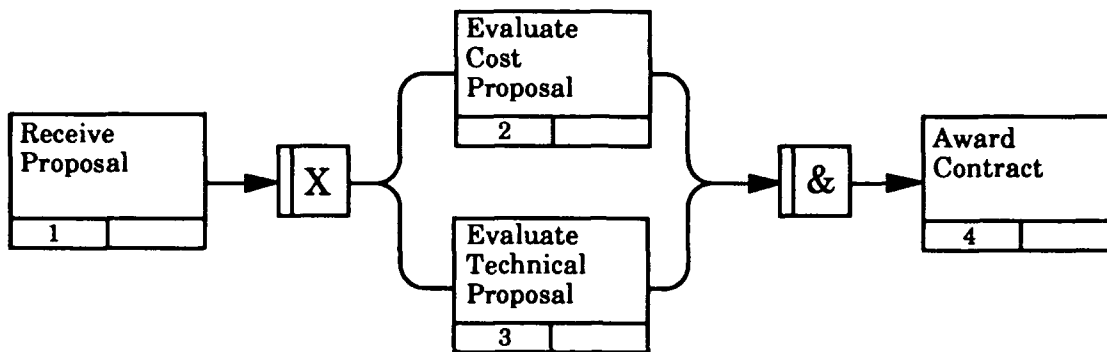


Figure 7-2
Invalid XOR/AND Structure Example

In Figure 7-2, after the UOB *Receive Proposal*, an XOR junction leads to two UOBs. This indicates that only one UOB--either *Evaluate Cost Proposal*--or *Evaluate Technical Proposal*, will be realized on any given instantiation of the diagram. Consequently, the UOB *Award Contract* could never be realized because the requirement that both UOBs preceding the AND junction be realized in the same activation can never be met. Why would anyone attempt to construct an IDEF3 diagram of this nature? Often, the real-world situation being described may have an undetected inconsistency that is a cause of concern for management. In the situation described previously, perhaps contracts were never awarded; thus, the IDEF3 diagram identified an organizational problem and enabled conflict resolution. A person creating an AS-IS description of a process may find situations of this type in an organizational diagram. Thus, it could be a semantically valid diagram of a situation, even if it is not model theoretically valid within the method. This type of structure is never correct in a TO-BE description of some proposed system, organization structure, or process. In either case, the description validation process should identify structures of this type as IDEF3 diagram errors.

7.2.2 Multiple Precedence Links Emerging from a Unit of Behavior Box

Consider the painting shop situation described by the IDEF3 diagram in Figure 7-3. Parts, after painting and drying, are subject to a quality check. If the test results indicate more paint is needed, the part is rerouted through the shop. Otherwise, the part leaves the shop. The diagram in Figure 7-3 is model theoretically incorrect because of the semantic ambiguity associated with the branching occurring out of the *Test Coverage* UOB.

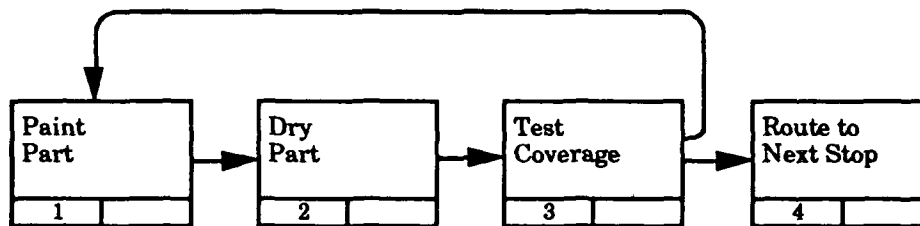


Figure 7-3

Example of an Ambiguous Branch in an IDEF3 Diagram

The fact that only one of the two branches emerging from the UOB will be taken is not captured by the topology drawn in the diagram. The solution to this problem is to acquire the additional facts needed to resolve the ambiguity. These facts may result in the addition of an XOR junction and a modification to the diagram as shown in Figure 7-4.

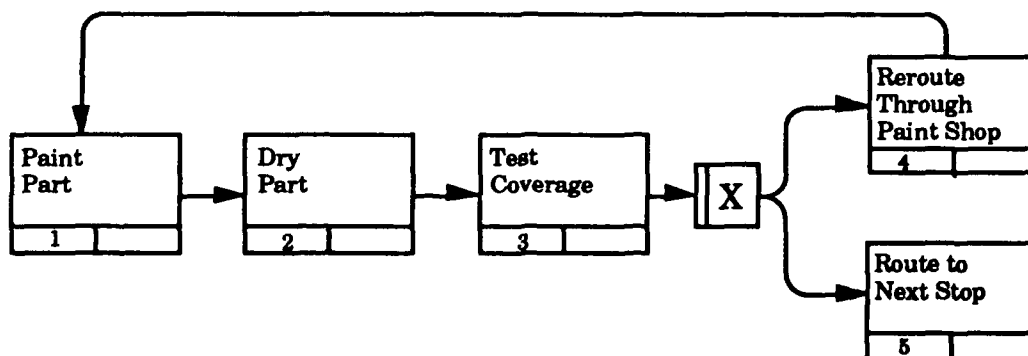


Figure 7-4

The Corrected Paint Shop IDEF3 Diagram

7.2.3 Multiple Leftmost Points for a Scenario or a Decomposition

An example of a scenario with multiple leftmost points is shown in Figure 7-5.

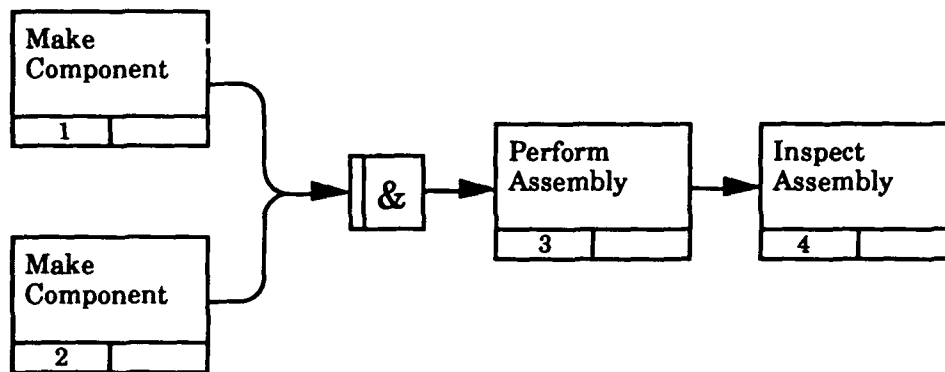


Figure 7-5
An IDEF3 Scenario With Multiple Leftmost Points

The solution to this commonly occurring model theoretic error is to add an appropriate junction box to the left of the diagram, as illustrated in Figure 7-6. In this example, the correct junction is a fan-out AND junction.

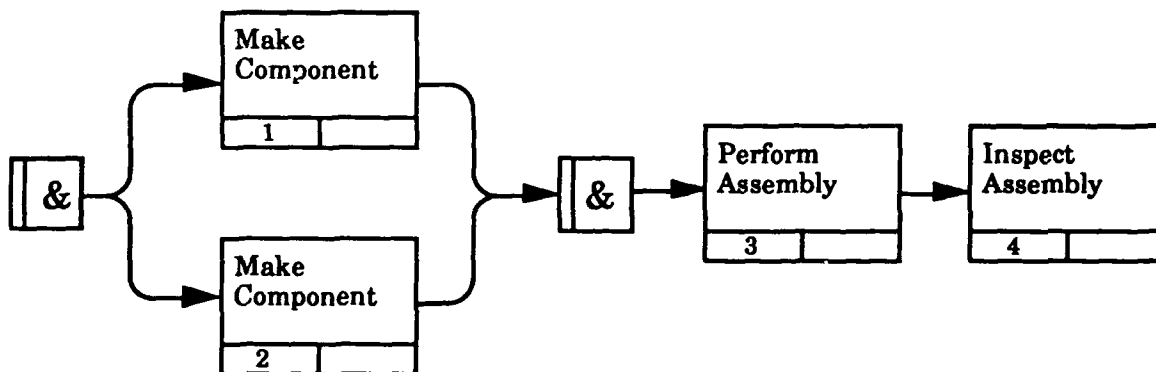


Figure 7-6
Correct IDEF3 Diagram for the Assembly Shop Scenario

Bibliography

- Allen J. F., (1984). Towards a general theory of action and time. Artificial Intelligence, 23, 123-154.
- Balci O., & Nance, R. E., (1987). Simulation model development environments: A research prototype. Journal of the Operational Research Society, 38(8), 753-763.
- Barwise, J. & Perry, J., (1983). Situations and attitudes. Cambridge, MA: MIT Press.
- Coleman, D. S. (1989). A Framework for Characterizing the Methods and Tools of an Integrated System Engineering Methodology (ISEM), Draft 2 Rev. 0. Santa Monica, CA. Pacific Information Management, Inc.
- Cross, S. (1983). Qualitative reasoning in an expert system framework, (T-124). Urbana, IL: University of Illinois Coordinated Science Lab.
- Cullinane, T., McCollom, N., Duran, P., and Thornhill, D. (1990). The human elements of IDEF. Unpublished manuscript.
- De Kleer J., (1979). Causal and teleological reasoning in circuit recognition, (TR-529). Cambridge, MA: MIT AI Lab.
- De Kleer J., & Brown J. S., (1984). A qualitative physics based on confluences. Artificial Intelligence, 24, 7-83.
- Devlin, K., (1991). Logic and information, volume I: Situation theory. Cambridge, MA: Cambridge University Press.
- Forbus K., (1984). Qualitative Process Theory. Artificial Intelligence, 24, 85-168.
- Formal foundations for an ontology description method, (KBSI-SBONT-91-TR-01-1291-02). (1991). College Station, TX: Knowledge Based Systems Inc.
- IDEF: Framework, draft report (IDEF-U.S.-0001). (1990). Dayton, OH: IDEF Users Group.
- IDEF4 Method Report, (1992). College Station, TX: Knowledge Based Systems, Inc.

IDEF5 Method Report, (1991). College Station, TX: Knowledge Based Systems Inc.

Information processing systems: Concepts and terminology for the conceptual schema and the information base (ISO/TR 9007). (1987). International Standards Organization.

Integrated computer-aided manufacturing (ICAM) architecture. Pt. II, Vol. IV: Function modeling manual (IDEF0) (DTIC-B062457). (1981). SofTech, Inc.

Integrated computer-aided manufacturing (ICAM) architecture. Pt. II, Vol. V: Information modeling manual (IDEF1) (DTIC-B062457). (1981). SofTech, Inc.

Integrated computer-aided manufacturing (ICAM) architecture. Pt. II, Vol. VI: Dynamics modeling manual (IDEF2) (DTIC-B062457). (1981). SofTech, Inc.

Integrated information support system (IISS). Volume 5: Common data model subsystem: Part 4: Information modeling manual (DTIC-A181952). (1985). General Electric.

Knowledge based information model integration, (1991). College Station, TX: Knowledge Based Systems Inc.

Kripke, S., (1963). Semantical considerations on modal logic. Acta Philosophica Fennica, 16, 39-48.

Lin, M. J., (1990). Automatic simulation model design from a situation theory based manufacturing system description. Unpublished doctoral dissertation, Texas A & M University.

Link, G. (1983). The logical analysis of plurals and mass terms: A lattice theoretic approach. In R. Bauerle (Eds.), Meaning, use, and interpretation. Berlin: De Gruyter.

Mayer, R. J., et al., (1987). Knowledge-based integrated information systems development methodologies plan, (Vol. 2) (DTIC-A195851).

Mayer R. J., (1988). Cognitive skills in modeling and simulation. Unpublished doctoral dissertation, Texas A&M University.

Mayer, R. J. (Ed.). (1990). IDEF0 function modeling: A reconstruction of the original Air Force report. College Station, TX: Knowledge Based Systems Inc.

- Mayer, R. J. (Ed.). (1990). IDEF1 information modeling: A reconstruction of the original Air Force report. College Station, TX: Knowledge Based Systems, Inc.
- Mayer, R. J. (Ed.). (1990). IDEF1X data modeling: A reconstruction of the original Air Force report. College Station, TX: Knowledge Based Systems, Inc.
- Mayer, R. J., Menzel, C. P., and deWitte, P. S. D. (1991). IDEF3 technical report. WPAFB, OH: AL/HRGA. Mayer, R. J., Menzel, C. P., and Mayer, P. S. D., (1991). IDEF3: A methodology for process description. WPAFB, OH: AL/HRGA.
- Mayer, R. J., Edwards, D. A., Decker, L. P., and Ackley, K. A. (1991). IDEF4 technical report. WPAFB, OH: AL/HRGA.
- Mayer, R. J., deWitte, P., Griffith, P., and Menzel, C. P. (1991). IDEF6 concept report. WPAFB, OH: AL/HRGA.
- Mayer, R. J., & deWitte, P. (1991). Framework research report. WPAFB, OH: AL/HRGA.
- Mayer, R. J., & Painter, M. (1991). IDEF family of methods. College Station, TX: Knowledge Based Systems, Inc.
- Mayer, R. J., & Wells, M. S. (1991). Integrated development support environment (IDSE) concepts and standards. WPAFB, OH: AL/HRGA.
- Menzel, C. P., Mayer, R. J., and Edwards, D. E., (1991). IDEF3 process descriptions and their semantics. In A. Kuziak & C. Dagli (Eds.), Knowledge base systems in design and manufacturing. Chapman Publishing.
- Menzel, C. P., & Mayer, R. J. (1990). IDEF3 formalization report. WPAFB, OH: AL/HRGA.
- The nature of ontological knowledge: A manufacturing systems perspective (KBSI-SBONT-91-TR-01-129-01). (1991). College Station, TX: Knowledge Based Systems, Inc.
- Ontology acquisition method requirements document (KBSI-SBONT-91-TR-01-1291-03). (1991). College Station, TX: Knowledge Based Systems, Inc.
- Ontology capture tool requirements document (KBSI-SBONT-91-TR-01-1291-04). (1991). College Station, TX: Knowledge Based Systems Inc.

- Ontology capture tool: Object-oriented design document (KBSI-SBONT-91-TR-0292-01). (1992). College Station, TX: Knowledge Based Systems, Inc.
- Overstreet C. M., & Nance R. E. (1985). A specification language to assist in analysis of discrete event simulation models. Communications of the ACM, 28(2), 190-201.
- Painter, M. K. (1990). Modeling with an IDEF perspective: Some practical insights. Proceedings, SME Autofact 90. Dearborn, MI: Society of Manufacturing Engineers.
- Pegden C. D. (1982). Introduction to SIMAN. State College, PA: Systems Modeling Corporation.
- Poole, D. (1990). A methodology for using a default and abductive reasoning system. International Journal of Intelligent Systems, 5, 521-548.
- Pritsker A. A. B., & Pegden C. D. (1979). Introduction to simulation and SLAM. New York, NY: Halsted Press.
- Reliable object based architecture for intelligent controllers. (1991). College Station, TX: Knowledge Based Systems, Inc.
- Ross, D. T. (1985). SADT today: A retrospective on an idea. IEEE Computer Magazine (special issue on Requirements Engineering).
- Soley, R. M. (Ed.). (1990). Object management architecture guide. Farmington, MA: Object Management Group, Inc.
- Tarski, A., The concept of truth in formalized languages. In Logic, semantics, and metamathematics. Oxford: Oxford University Press.
- Zachman, J. (1987). A framework for information systems architecture, IBM Systems Journal, 26(3), 276-292.

Index

- Activation of a Diagram 47, 50
- Constraints (in IDEF3 Descriptions) 22, 24, 32, 56, 61, 74, 100, 108
- Decomposition, *see* UOB
 - Decomposition*
- Description
 - Definition 1
- Facts (in IDEF3 Descriptions) 4, 22, 24, 32, 74, 82, 100, 108
- Glossary of an IDEF3 Element 24, 33, 59, 62, 80, 81
- IDEF xi, xv
- IDEF1 66, 70, 79, 81
- IDEF1X 81
- IDEF3 Description xv, 4, 5, 10, 19, 21, 22, 70
 - AS-IS Description 122, 125
 - Data Capture 69
 - Description Reading Process 115
 - IDEF3 Pools 74
 - Interview 70, 98
 - Type of Information to be Gathered 72
 - Model Theoretic Validation 121, 122, 125
 - Object Centered View 14, 19, 56, 79
 - Object Pool 73, 74, 99
 - Object State Pool 74
 - Partial Description 29, 81
 - Process Centered View 11, 19
 - Process Flow Description 10
 - Process Flow Diagram 4, 5, 11, 16, 21, 41, 54, 101, 104
 - Layout 30
 - Viewpoint 117
 - Process Flow Path 6, 15, 34, 35, 37, 38, 48, 51, 54, 101
 - Purposes 4
 - Scenario Pool 74
 - Semantic Validation 121, 122
 - Situation Descriptions 73
 - Source Material Log 70, 71
 - Statement of Context 67
 - Statement of Need 68
 - Statement of Purpose 67, 69, 117
 - Statement of Scope 68, 69, 117
 - Syntactic Validation 121, 122
 - The Big Picture 117
 - TO-BE Description 122, 125
 - UOB Pool 73, 74, 100
 - Validation 121
- IDEF3 Description Development 65
 - Project Team 65
 - Analyst 65, 72, 74, 80, 97

- Client 65, 67, 97
- Domain Expert 65, 69, 72, 82, 97
- Primary Contact 65, 70
- Project Leader 65
- Reviewers 65
- Team Member 66
- IDEF3 Method
 - Application Area
 - Business Systems Modeling 2
 - Design Data Life-Cycle 3
 - Project Management
 - Support 3
 - System Requirements
 - Descriptions 3
- IDEF3 Process Description
 - Capture Method xviii, 1
- IDEF5 66
- IDEFØ xii, 4, 66, 70, 78
- Junction 6, 13, 34, 35, 37, 38, 40, 49, 51, 53, 56, 76, 111, 116, 120, 122, 123, 127
 - "&" 13
 - AND Junction 13, 34, 35, 37, 124
 - Asynchronous 38, 50
 - Fan-in 37, 38, 125
 - Fan-out 37, 54
 - Synchronous 38, 39, 51
 - Asynchronous 38
 - Fan-in 37, 38, 50
 - Fan-out 38, 50
- Classifications 34
- Fan-in Junction 21, 32, 34, 35, 37, 50, 120, 123, 124
- Fan-out Junction 21, 32, 34, 35, 36, 37, 54, 56, 101, 120, 123, 124
- Junction Numbers 41
- OR Junction 34, 35, 54, 124
 - Asynchronous 51
 - Fan-in 38, 52, 54
 - Fan-out 52
 - Synchronous 52
- Semantics 37
- Synchronous 38
 - Fan-in 37
 - Fan-out 38
- XOR Junction 15, 34, 35, 38, 101, 111, 124
 - Fan-out 38, 124, 125
- Kit 82, 83
- IDEF3 Kit Cover Sheet 89
- Kit Production 82
- Kit Review 82
 - Commentor Guidelines 86
 - Description Kit 86
 - Diagram Form 92
 - Kit Contents 88
 - Kits
 - Commentor/Analyst Meeting 88
 - Notes, Author/Commentor 87
 - Object Kit 86
 - Review Procedure 84
 - Roles

- Analyst 83, 85
- Commentor 83, 85
- Librarian 83
- Readers 83
- Reviewers 83
- Scenario Kit 84, 85, 86
- Node Index/Contents Sheet 94
- Notes Field 94
- Number Field 95
- Standard Kit 88
- LDFD xii
- Link 6, 12, 21, 28, 29, 30, 32, 37, 47, 49, 123, 124
 - Link Destination 32
 - Link Identification Numbers 34
 - Link Rules 123
 - Link Source 32
 - Link Specification 77, 86
 - Link Specification Document 30, 31, 32, 48, 112, 119
 - Constraints 32
 - Facts 32
 - Objects 32
 - Link Type
 - Dashed Link 21
 - Object Flow Link 21, 30, 31, 49
 - Precedence Link 12, 21, 30, 31, 48, 51, 123, 124
 - Relational Link 30, 48
 - User Defined Link, *see Relational Link*
 - Links out of Junctions 36
- Method xii, xiv
- Methodology xii
- Model xiv, xviii
 - Definition 1
- Object (Element in an IDEF3 Description) 1, 2, 3, 4, 6, 9, 10, 11, 12, 13, 14, 24, 62, 72, 79, 99
- Object Centered View (*see also* OSTN Diagram) 6, 10, 14, 112
- Object State (The State of an IDEF3 Object) 16, 58, 59, 79, 80
 - Entry Conditions 15, 60, 62
 - Exit Conditions 15, 60, 62
 - Object State Description Form (OSD) 17, 60, 61, 62, 81
 - State Description 60, 62
- Object State Transition Network (OSTN) 4, 6, 10, 13, 14, 59, 80, 86
 - Diagram 62, 79, 80, 81
 - Focus Object 59
 - OSTN Network Description Form 58, 59
 - State Transition 6
- Object vs. Object State 58
- OSD, *see Object State Description Form (OSD)*
- OSTN, *see Object State Transition Network (OSTN)*
- Process Centered View (*also* Process Flow Diagram) 10, 11
- Process Flow Diagram
 - Structures 119, 123, 124
- Process Flow Diagrams, *see Process Flow Diagram in IDEF3 Description*

Process Flow Network 11
 Referent 6, 12, 13, 17, 21, 41, 45, 54, 108
 Asynchronous 44, 62
 Attached to Transition Arc 63
 Identifier 43
 in an OSTN 62
 Referent Symbol Structure 43
 Referent Symbol Syntax 41
 Asynchronous 42
 Referent Type
 Elaboration 44, 46, 56, 110
 Go-To 41, 43, 46, 54
 Junction 41, 54
 Note 44
 Object 41, 44, 53, 56
 OSTN 41, 44, 45, 62
 Scenario 44, 45, 58, 62
 UOB 41, 43, 44, 46, 55, 62
 Synchronous 42, 45, 62
 Unconditional 41, 43
 Scenario 6, 9, 10, 67, 69, 115, 122, 127
 Stand-alone UOB 29
 Status Field 92
 Teamwork, Disciplined 83
 Tool xii
 Transition Arc 62
 Unit of Behavior (UOB) 6, 21, 22, 25, 27, 30, 31, 47, 48, 77, 78, 100
 Activation 37
 Constraints 24
 Elaboration 22, 25
 Elaboration Document 22
 Elaboration Document
 Description of Contents 23
 Facts 24
 Instance 47
 Objects 24
 Realization 47
 UOB Box 22, 28, 78
 UOB Elaboration 13, 21, 22, 32, 77, 86, 101, 104, 123
 UOB Elaboration Document 23
 UOB Numbering 27
 UOB Decomposition 13, 14, 21, 22, 25, 27, 77, 85, 117, 122, 123
 Definition 25
 Objective View 25
 Parent UOB 25
 Role View 25
 UOB Instance 47
 UOB Realization 47
 UOB, *see Unit of Behavior (UOB)*